

RAPID PROTOTYPING AND EVALUATION OF DIALOGUE SYSTEMS FOR
VIRTUAL HUMANS

by

Sudeep Gandhe

Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(COMPUTER SCIENCE)

July 2012

Copyright 2014

Sudeep Gandhe

Table of Contents

List Of Tables	iv
List Of Figures	v
Abstract	ix
Chapter 1 Introduction	1
1.1 Dialogue Systems and Virtual Humans	1
1.2 Dialogue System Architectures	5
1.3 Building a Dialogue System	8
1.4 Evaluating a Dialogue System	10
1.5 Bottlenecks in Dialogue System Development	13
1.6 Trade-offs in Dialogue System Development	15
1.7 Thesis Statement	15
1.8 Thesis Contributions	16
1.9 Outline	19
Chapter 2 Related Work	21
2.1 Authoring Tools for Dialogue Systems	21
2.2 Data-driven approaches for Dialogue Modeling	22
2.3 Evaluating Dialogue Models	25
Chapter 3 Rapid Development of Advanced Question Answering characters by Non-experts	27
3.1 Advanced Question Answering Characters	28
3.2 Dialogue System Architecture	28
3.3 Dialogue System Authoring	30
3.4 DomainEditor: Integrated Authoring tool	31
3.4.1 Domain Knowledge Level	32
3.4.2 Dialogue Act Level	33
3.4.3 Surface Text Level	35
3.5 Dialogue Manager	37
3.5.1 Functions of Dialogue Manager	38
3.5.2 Policy Authoring	40
3.6 Evaluation	41
3.6.1 Evaluation of the Dialogue Act Scheme	42
3.6.2 Evaluation of the Integrated Authoring Tool, DomainEditor	44
3.6.3 Resulting Virtual Human Dialogue Systems	48

3.7	Conclusion	49
Chapter 4	Unsupervised Dialogue Models	54
4.1	SASO-ST Testbed	54
4.2	Dialogue System Architecture	55
4.3	Formulating a Response: Generation Vs Selection	57
4.4	Viability of the Selection Approach	58
4.5	Selection Criterion	60
4.6	Unsupervised Dialogue Models	62
4.6.1	Random	63
4.6.2	Nearest Context	64
4.6.3	Segmented Nearest Context	65
4.6.4	Segmented Random	68
4.6.5	Cross-lingual Relevance Model	68
4.6.6	Perceptron	70
4.7	Evaluation of Dialogue Models	74
4.7.1	Dynamic Context	75
4.7.1.1	Dynamic Context Evaluation by Dialogue Participants	75
4.7.1.2	Dynamic Context Evaluation by Bystanders	76
4.7.2	Static Context	79
4.7.2.1	Wizard Data Collection	79
4.7.2.2	Comparative Evaluation of Models	82
4.8	Discussion	84
4.9	Applications	86
4.10	Conclusion	88
Chapter 5	Automatic Evaluation for Dialogue Models	89
5.1	Static Context Evaluation	90
5.1.1	Weak Agreement	90
5.1.2	Voted Appropriateness	91
5.1.3	Discussion	92
5.2	Dynamic Context Evaluation	92
5.2.1	Information Ordering	93
5.2.2	Evaluating Information Ordering	94
5.2.3	Experimental Setup	95
5.2.4	Viability of Information Ordering Task and Human-level Upper Baseline	97
5.2.5	Automatic Evaluation for the Information Ordering Task	98
5.2.5.1	Holistic Evaluation	98
5.2.5.2	Turn-by-Turn Evaluation	99
5.2.6	Discussion	100
5.3	Conclusion	102
Chapter 6	Conclusion	103
6.1	Summary	103
6.2	Future Work	104
	Bibliography	107

List Of Tables

1.1	Three types of dialogue systems.	4
3.1	A summary of dialogue act annotation reliability and domain coverage for different corpora at different developmental stages for Amani.	42
3.2	Subjective evaluation for Hassan with old and new architectures and corresponding authoring processes	48
3.3	Evaluation of the resulting Virtual Human dialogue systems for characters built using our authoring process and the integrated authoring tool, DomainEditor.	49
4.1	Corpus details for different domains, expected value of $maxsim_f$ scores along with percentage of utterances with exact or approximate match.	60
4.2	Results for the four dialogue models as evaluated by the dialogue participants themselves in dynamic context setting.	76
4.3	Results for the four dialogue models as evaluated by judges not participating in the conversation (bystanders) in dynamic context setting.	77
4.4	Results for two dialogue models in embodied character settings and speech as input modality along with the evaluation of human-human dialogues performed by bystanders in dynamic context setting.	77
4.5	Overlap measures indicating inter-wizard agreement	81
4.6	Offline comparative evaluation of dialogue models.	84
4.7	Pairwise comparisons for evaluating appropriateness in static context setting along with statistical significance.	85
5.1	Examples of observed sequences and their respective b_2 , b_3 & τ values. Here the reference sequence is [0,1,2,3,4,5,6,7,8,9].	95

List Of Figures

1.1	Dialogue excerpts from different types of dialogue systems.	2
1.2	Dialogue Act based architecture of a spoken dialogue system and required resources . . .	6
1.3	Examples of dialogue acts from different domains	7
1.4	Surface Text based architecture of a spoken dialogue system and required resources . . .	8
1.5	Process of building a dialogue system.	9
1.6	A sample domain-independent information state update rule for SASO-ST virtual human dialogue system	10
1.7	A schematic representation of various decision factors in evaluating dialogue models for virtual humans	11
1.8	Schematic representation of <i>Dynamic Context</i> and <i>Static Context</i> evaluation settings. . .	11
1.9	A sample domain-dependent information state update rule for SASO-ST virtual human dialogue system	14
3.1	An example of advanced question answering dialogue with virtual human character, Hassan . <i>Captain</i> refers to the human trainee. Third column indicates dialogue act type. .	29
3.2	Architecture for the Tactical Questioning Dialogue System	30
3.3	DomainEditor: An Integrated Authoring tool for designing the domain, and specifying the utterances that map to various dialogue acts.	33
3.4	Aspects of the <i>Hassan</i> domain	34
3.5	Sample dialogue acts automatically generated from the <i>Hassan</i> domain along with example utterances.	35
3.6	A sample dialogue act along with the corresponding surface text utterances. The most salient part of these utterances which matches with the dialogue act is highlighted. . . .	37
3.7	State charts for <i>Hassan</i> domain.	38
3.8	Example dialogue excerpts showing topic tracking & grounding functions of the dialogue manager. The last column indicates the <i>topic</i> of conversation after processing the corresponding utterance.	39
3.9	The authoring tool can be used to specify the conditions for question-answering network. .	40
3.10	Example dialogue showing the currently active states for the networks in Figure 3.7. . . .	41
3.11	Various Virtual Human characters that have been created using DomainEditor.	45
3.11	Various Virtual Human characters that have been created using DomainEditor.	46
3.12	Amount of dialogue system resources collected across time for several characters which were authored using DomainEditor by non-experts.	47
3.13	A sample dialogue with Amani	50
3.14	A sample dialogue with PFC Sean Avery character. <i>Player</i> refers to the human trainee. Third column indicates dialogue act type.	51
3.15	Types of dialogue acts and the dialogue behaviors supported by them.	52
3.15	Types of dialogue acts and the dialogue behaviors supported by them.	53
4.1	Virtual Human for Doctor character from SASO-ST scenario.	55

4.2	A sample role-play dialogue in SASO-ST domain.	56
4.3	A sample Wizard-of-Oz dialogue in SASO-ST domain.	56
4.4	An example illustrating the contrast between <i>Generation</i> and <i>Selection</i> approach.	58
4.5	Expected value of $maxsim_{Meteor}$ vs # utterances in the training data for different domains.	61
4.6	Expected value of $maxsim_{Meteor}$ vs # utterances in the training data for SASO-ST domain.	61
4.7	A schematic representation of implemented unsupervised dialogue models and the relationships between the information used by their ranking functions.	63
4.8	Example interaction for <i>Random</i> in dynamic context setting.	64
4.9	Feature vector representing the context $\langle u_{t-2}, u_{t-1} \rangle$ of previous $n = 2$ turns.	65
4.10	Example interaction for <i>Nearest Context</i> in dynamic context setting.	66
4.11	List of key concepts along with the representative unigrams compiled for SASO-ST dialogues.	67
4.12	Example interaction for <i>Segmented Nearest Context</i> in dynamic context setting.	67
4.13	Example interaction for <i>Segmented Random</i> model in dynamic context setting.	68
4.14	Example interaction for <i>Cross-lingual Relevance Model</i> model in static context setting.	70
4.15	Features extracted from a context ($context_j$) and a response utterance (u_i)	73
4.16	Example interaction for <i>Perceptron</i> model in static context setting.	74
4.17	A screen-shot of the user interface for dynamic context evaluation by the dialogue participants.	75
4.18	Average appropriateness levels for the four dialogue models evaluated by the dialogue participants themselves in dynamic context setting.	76
4.19	Average appropriateness levels for various dialogue models.	78
4.20	Scatter plot of average appropriateness levels for different models as judged by two judges.	78
4.21	A screenshot of the interface for the wizard data collection.	80
4.22	A Histogram for the number of selected appropriate response utterances.	81
4.23	Avg. cardinality of the set U_c^R – union of sets of utterances selected as appropriate responses by wizards R for different values of $ R $	82
4.24	Screenshot of the user interface for static context comparative evaluation of dialogue models	83
4.25	Results of static context comparative evaluation of dialogue models.	84
4.26	Illustration of the problem due to the granularity of the utterance. This dialogue is generated by using <i>Segmented Nearest Context</i> model and shows the evaluation by the dialogue participant in dynamic context setting. Last utterance from the doctor gets a low rating.	87
5.1	Appropriateness of responses (R) as judges by 4 human judges plotted against the number of wizard votes (V) received by those responses. The dashed line indicates a fitted linear model. A small amount of jitter is added to V for visualization.	91
5.2	Comparison between two automatic evaluation understudy measures at system level in static context setting.	92
5.3	A random permutation of a human-human dialogue [Source: (Bratt et al., 1995)]. The original reference order can be recovered by reading the dialogue in sequence $\langle 6,9,2,3,4,7,8,5,10,1 \rangle$	94
5.4	A segment from a negotiation role-play dialogue.	96
5.5	A segment from a travel agent dialogue.	96
5.6	A segment from a television show dialogue. [source: http://www.twiztv.com/]	97
5.7	Human-level upper baseline for information ordering task (human performance)	98

5.8	Single coherence rating per permutation.	99
5.9	Screenshot of the interface used for collecting coherence rating for dialogue permutations.	100
5.10	Turn-by-turn coherence rating per permutation	101
5.11	Distributions for Kendall's τ , $(b_2 + b_3)/2$ and the relationship between them for all possible dialogue permutations with 10 turns and earlier mentioned constraints.	101

List of Algorithms

3.1	Algorithm for generation of dialogue acts from domain specification	36
4.1	Perceptron Training Algorithm	71

Abstract

This thesis presents contributions towards *Rapid Prototyping and Evaluation of Dialogue Systems for Virtual Humans*. Different architectures have been proposed for developing Virtual Human Dialogue Systems. These can be broadly classified in two categories based on the level at which dialogue management occurs – *Dialogue Act* level and *Surface Text* level. This thesis makes contributions for both the types of architectures.

For *Dialogue Act based* architectures, collecting the required resources is costly, time-consuming and requires expertise in dialogue system development. **The first contribution of the thesis is an authoring process designed for dialogue act based architectures and the genre of Advanced Question-Answering dialogues which allows non-experts to author the required resources rapidly.** We demonstrate its viability by implementing the necessary integrated authoring tool and having *non-experts* build Advanced Question-Answering Virtual Humans. Our authoring process and the accompanying tool allows *non-experts* to build systems faster (within a few weeks) compared to what experts used to be able to do without the tool (up to several months).

For *Surface text based* architectures, this thesis addresses two major challenges: the need for models that can combine arbitrary information state annotations with the surface text corpus and rapid, cost-efficient evaluation of the resulting dialogue models. We compare two approaches for formulating a response for surface text based dialogue models: *Generation* and *Selection*. **As a second contribution of the thesis, for the first time in literature, we propose an empirical method to determine whether the selection approach is viable for a given domain and apply it to 10 different domains.** It shows that for some domains and corpora, the *selection* approach is viable where an acceptable percentage of utterances are the same or substantially similar to already seen utterances.

Surface text based architectures require relatively low-cost resources such as dialogue transcripts. But without high-level information state representations such dialogue systems cannot adequately model complex behaviors. To better understand the trade-offs between the cost of building a specific set of resources and the performance of the resulting dialogue system, we need flexible dialogue system architectures that allow novel combinations of resources. **As a third contribution of this thesis, we develop flexible architectures that allow novel combinations of different types of resources, such as surface text transcripts and information state annotations, and systematically evaluate them in three different evaluation settings.** We implemented and evaluated 8 types of models and demonstrate the relative utility of different resource combinations and architectures.

For *Surface text based* architectures, evaluating the performance of the resulting dialogue system involves collecting subjective judgments about the appropriateness of responses given the dialogue context. Since the evaluation process requires a lot of human participation, it is time-consuming and costly. Our approach towards reducing the cost of estimating the performance of the dialogue system is to reduce the human involvement in evaluation. **As a final contribution, we have evaluated two previously proposed automatic evaluation measures in terms of how well they correlate with human judgments and have developed two new automatic measures, that achieve higher correlation with human judgments.**

Chapter 1

Introduction

Dialogue can be defined as an exchange of contributions between more than one participant. Each new contribution is coherent with the previous contributions and accumulates into a dialogue through an interactive process. Two friends chatting, a job interview, a debate, an email thread, a user interacting with a GUI can all be considered examples of dialogue. The contributions in a dialogue can come in the form of different modalities - speech, typed text, non-verbal gestures like head-nod, gaze, hand movements, GUI displays, mouse clicks, sketching etc. Different modalities can be used by different participants often in a complementary manner. In this dissertation, we focus on highly interactive dialogues like those in a conversational setting rather than less interactive forms (e.g., email).

In this introductory chapter, we first look at what dialogue systems are, specifically virtual human dialogue systems. We compare and contrast virtual human dialogue systems with other kinds of dialogue systems. Next we describe the typical process involved in building and evaluating such a dialogue system. We identify the bottlenecks involved in dialogue system development and the trade-offs between the cost of collecting required resources and the benefits of the resulting dialogue system. Next we present the thesis statement followed by the thesis contributions and how each contribution addresses the bottlenecks presented earlier. We conclude this chapter with an outline of the rest of the dissertation.

1.1 Dialogue Systems and Virtual Humans

Dialogue systems are computer programs that can interact with humans or other agents using natural language. Over the years many dialogue systems have been built addressing a wide range of goals; Some help users complete a specific task (e.g., Communicator – a travel planning dialogue system (Walker et al., 2001) and Let's go – a bus schedule information providing system (Raux et al., 2005)); some are designed to act like humans (e.g., Virtual Humans, that can be used for simulation training (Traum et al., 2008c)); some are designed to be conversational partners (e.g., chatbots (Weizenbaum, 1966)); some are designed for tutoring and can engage in a conversation with students to achieve certain learning objectives (e.g., (Litman and Silliman, 2004)). See Figure 1.1 for some example dialogues from different types of systems. In these examples the dialogue system actively participates in the dialogue. There are also cases where dialogue systems take part in a conversation only passively. For example, in a Speech-to-Speech translation system (e.g., Verbmobil (Kay et al., 1994), Transonics (Narayanan et al., 2004)), the dialogue system only mediates the dialogue between speakers of different languages.

1	User	yes
2	System	what is your full name?
3	User	adam simons
4	System	hello, adam simons, i'm glad to hear from you again. you can ask me for help at any time. what city are you leaving from?
5	User	hartford connecticut
6	System	a flight from hartford. where would you like to go?
7	User	orlando florida
8	System	traveling to orlando. on what date would you like to travel
9	User	october thirty first late morning
10	System	traveling late morning. on what date would you like to travel

(a) Dialogue excerpt from a DARPA communicator dialogue system, a task oriented system for travel planning (Prasad and Walker, 2004)

1	Captain	Hello Doctor Perez.
2	Doctor	Hello.
3	Captain	Very nice to meet you!
4	Doctor	What do you want?
5	Captain	I have orders to assist you in moving this clinic to a safer location.
6	Doctor	You want to move the clinic?
7	Captain	Yes
8.1	Doctor	This conflict is madness, it is killing people!
8.2	Doctor	We need proper supplies here!
9	Captain	It is not safe here we can't protect you.
10.1	Doctor	Protect me? Protect me from what?
10.2	Doctor	Are you going to attack?

(b) Dialogue excerpt from SASO-ST (Traum et al., 2005), a virtual human dialogue system used for training cross-cultural negotiation skills. The trainee captain has to convince a virtual human doctor to move his clinic from its current location.

Figure 1.1: Dialogue excerpts from different types of dialogue systems.

Virtual Humans

Virtual Humans (VH) are autonomous agents who can play the role of humans in simulations (Traum et al., 2008c). These agents have a human-like animated body and are typically situated in a virtual world. They can perceive the state of this virtual world and can manipulate it through certain actions. They can interact with real humans or other virtual humans through natural language. Apart from the verbal signals (speech or typed text), these virtual humans can also communicate using non-verbal signals (e.g., body posture, facial expressions, hand gestures).

Virtual Human characters have proved useful in many fields; some have been used in simulations for training negotiation skills (Traum et al., 2005) or tactical questioning skills (Traum et al., 2008a); some virtual humans are used in settings where a face-to-face conversational setting has a stronger impact in presenting some information (e.g., a Virtual Nurse used for counseling hospital patients who have inadequate health literacy at the time of discharge (Bickmore et al., 2009), Museum Docents promoting science and technology interests in middle school students (Swartout et al., 2010), a virtual soldier used during U.S. Army recruiting events (Artstein et al., 2008)); some virtual humans are used as non-playing

characters in interactive games (e.g., (Gustafson et al., 2004)). Although these different virtual humans may have different sets of goals, one common goal shared by them all is to produce *believable human-like* behavior. An ability to take part in conversations using natural language is important for believable virtual humans. This interface has to be good enough to engage the trainee or the gamer in the activity.

Virtual Human Dialogue Systems

Virtual humans with an ability to act as a human-like conversational partner will have a dialogue system associated with them. Here, we refer to this dialogue system as a Virtual Human Dialogue System. Although virtual humans routinely use non-verbal signals for effective communication, in this work we focus only on the verbal interaction.

Natural language dialogue systems can be compared across different criteria such as the goal of the dialogue system, how it is evaluated, what level of understanding is required, what metaphor the dialogue system projects, etc. Table 1.1 summarizes different types of dialogue systems and how virtual human dialogue systems compare to others.

Chatbot systems like Eliza (Weizenbaum, 1966) or Alice (Wallace, 2003b) aim to imitate humans. These are conversational systems which have a goal to be human-like and have to operate in an unrestricted domain. The user utterances can be about any topic the user can think of. On the other hand, task-oriented dialogue systems such as Communicator (Walker et al., 2001), ATIS (Seneff et al., 1991) or Trains (Allen et al., 1995) restrict the user quite severely in the allowed topics and ways of talking about them. Compared to these, virtual human dialogue systems fall somewhere in between. Similar to chatbots, virtual humans need to be human-like conversational partners, but typically their interactions are situated within a background narrative. This restricts the domain of conversation to a specific scenario.

The level of understanding required is different in different dialogue systems. For chatbots, it is often sufficient to talk about topics at a fairly shallow level, without requiring a lot of detailed task knowledge or knowledge of how some parts of a task relate to others. This luxury is not available for a task oriented dialogue system where the system is expected to complete a task or provide task-specific information. There are some domains that fall between these extremes, for instance negotiation about whether or not to adopt a proposal. In this case, there is definitely a task or set of tasks involved, but one does not necessarily require as detailed knowledge as is required to actually perform the task.

Another dimension along which dialogue systems can be differentiated is how its developers view the system – as an *interface* to some task or as a *simulation* of natural language use in humans (Larsson, 2005). Edlund et al. (2008) point out that users of dialogue system perceive it metaphorically and this metaphor guides user's expectations from the system. There are two possible metaphors that a system can project – *interface* or *human*. Edlund et al. argue that a dialogue system needs to be careful about which metaphor it projects and remain internally coherent with it. Voice interface to television (e.g., (Ibrahim and Johansson, 2002)), in-car dialogue systems (e.g., CHAT (Weng et al., 2006), Ford Sync by Microsoft¹) and Communicator are examples of an *interface*, where speech replaces or augments the traditional interface mechanisms such as touchscreen, menus, buttons, etc. On the other hand, virtual human dialogue systems by definition need to project the *human* metaphor. This dimension also becomes relevant for evaluating the dialogue systems. For dialogue systems using the *interface* metaphor, suitable evaluation criteria include user satisfaction, task completion and dialogue efficiency. On the other hand, for dialogue systems using the *human* metaphor an important criterion would be human-likeness.

¹<http://http://www.ford.com/technology/sync>

	Chat-bots	Virtual Humans	Task Oriented
Domain	Unrestricted	Somewhat restricted	Restricted
Goal	Be human-like	Be human-like and complete task	Complete the task efficiently
Under- standing	Shallow or no understanding of progression of dialogue needed	Shallow understanding of dialogue progression needed	Deep understanding of dialogue progression needed
Operating Level	Surface text	Surface text or Dialogue Act	Dialogue Act
Method	Keyword-spotting, pattern matching, corpus based retrieval	Information-state based or corpus based retrieval	Information-state based, form based, finite state models
Metaphor	Human	Human	Mostly Interface, sometimes Human
Evaluation	Turn by turn appropriateness ratings, Turing test	Appropriateness of responses, Dialogue Coherence	User satisfaction, Task completion, Dialogue efficiency
E.g.	Eliza (Weizenbaum, 1966), Alice (Wallace, 2003b)	SASO-ST (Traum et al., 2005), Sgt Blackwell (Leuski et al., 2006), NICE (Gustafson et al., 2004)	Trains (Allen et al., 1995), Communicator (Walker et al., 2001), Let's Go (Raux et al., 2005)

Table 1.1: Three types of dialogue systems.

Dialogue Genres

The dialogues resulting from these systems can be classified into different *genres*. E.g.,

- *Information access*, where users are using spoken language to access some information such as weather (Zue et al., 2000) or bus schedules (Raux et al., 2005)
- *Transactions*, where users complete well defined tasks such as booking airlines as in Communicator (Walker et al., 2001) or reserving conference rooms (Bohus and Rudnicky, 2003)
- *Command-and-Control*, where users are issuing commands to control devices like TV or mobile phones
- *Question-Answering*, where users can interview the system to find information of interest as in visitors interviewing a virtual soldier about U.S. Army careers (Artstein et al., 2008)
- *Advanced Question-Answering*, where a response to a user's question may vary depending on a number of factors such as the dialogue history, the virtual character's social attitudes, its personality, whether the question is considered sensitive in that particular domain, etc. In some cases, the virtual character may answer the question cooperatively. In other cases, it may choose to be non-cooperative and withhold the answer or even lie. It may also choose to engage in bargaining behavior (Gandhe et al., 2008).
- *Negotiation*, where users can engage in a negotiation as in whether to move a clinic from its current location (Traum et al., 2005).

Although this is not an exhaustive list it serves the purpose of illustrating the variety of dialogue genres. Within each genre the *domain* of interaction can be different. E.g., Within the genre of information access weather and bus schedules are different domains. It is possible for a dialogue system to support more than a single dialogue genre, but typically most dialogue systems are designed for a specific dialogue genre. Some genres such as *Question-Answering* and *Negotiation* are more prevalent in virtual human dialogue systems than other genres such as *Command-and-Control*.

1.2 Dialogue System Architectures

Different architectures have been proposed for different types of dialogue systems. The specific architecture chosen is dictated by the specific goals of the dialogue system and its evaluation criteria. This architecture, in turn, determines the specific types of resources required. Dialogue system architectures can be crudely classified in two categories based on the level at which dialogue management occurs – *Dialogue Act* level and *Surface Text* level. Task-oriented dialogue systems typically use the dialogue act based architecture whereas chatbots are based on the surface text based architecture. Virtual human dialogue systems fall somewhere in between and can work at either the dialogue act level or the surface text level. E.g., the virtual human designed for simulation training of negotiation skills (Traum et al., 2005) operates at the dialogue act level, whereas the virtual soldier, SGT Star (Artstein et al., 2008), presented at U.S. Army events, operates at the surface text level based on corpus-based retrieval methods (Leuski et al., 2006).

Dialogue Act based architecture

Figure 1.2 represents an architecture where dialogue management happens at the dialogue act level and shows the resources required by such an architecture. *Dialogue acts (DA)* are a generalization of the speech act notion where every spoken utterance is considered as an action (Austin, 1962; Searle, 1969).

The speech signal is converted to a stream of word tokens by an Automatic Speech Recognizer (ASR). The output of the ASR is text and can optionally include an n-best list of hypotheses along with confidence scores. The ASR output is then passed over to a Natural Language Understanding

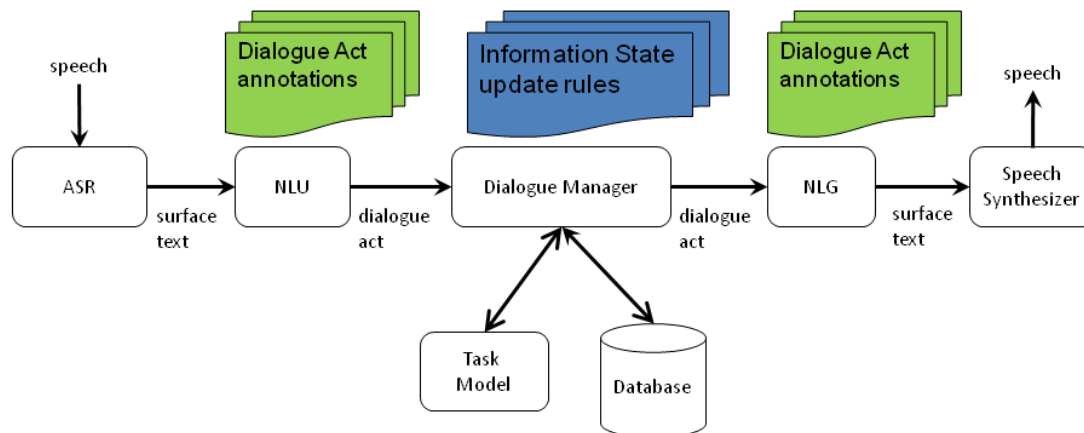


Figure 1.2: Dialogue Act based architecture of a spoken dialogue system and required resources

(NLU) module which maps the surface text to one or more dialogue acts along with corresponding semantic representations. The ASR can employ either statistical language models or context free grammars. If the ASR uses context free grammars augmented with semantics then the ASR output will contain the recognized text along with the associated semantics. In such a case a separate NLU module is not required.

The Dialogue Manager (DM) receives user dialogue act(s) as input and formulates system dialogue act(s) as a response to user's input or as a system initiative. The dialogue manager is responsible for maintaining the dialogue state. The dialogue manager updates the dialogue state upon receiving a user dialogue act or on sending a system dialogue act. Dialogue acts can be defined at multiple levels – core speech acts as well as other acts for turn-taking, grounding and argumentation (Traum, 1999). From the dialogue management perspective, the important aspects of core speech acts are the illocutionary force type (e.g., request, question, assertion, etc.) and the propositional content. This propositional content/semantics is often application-dependent as it is used to query databases and/or interact with application-specific task models. Figure 1.3 shows example utterances with the corresponding dialogue acts for different dialogue systems. From the perspective of dialogue acts, different *genres* of dialogues can be characterized by specific distributions of dialogue act sequences, specifically the illocutionary force type. Whereas, different *domains* will result in different propositional semantics.

The system dialogue act forms the input for Natural Language Generation (NLG), which generates the surface text for the given dialogue act. This text is finally converted to speech by a speech synthesis module. In case of virtual human dialogue systems, this synthesized speech is then performed by an animated body along with appropriate non-verbal behavior.

Task oriented dialogue systems typically use a dialogue act based architecture. Form-filling dialogue models (Goddeau et al., 1996), agenda-based dialogue models (Xu and Rudnicky, 2000) and information state dialogue models (Larsson and Traum, 2000) all use the dialogue act based architecture. Operating at the dialogue act level allows for easy integration with other kinds of knowledge-bases and reasoning, planning modules. But it comes with a price of more processing to translate from the surface text level to a higher level of abstraction (dialogue act) and back. Often, the NLU and NLG components employ statistical methods for converting between surface text

We will have to move the hospital.

speech-act	assert		
actor	captain		
addressee	doctor		
semantics	task	move-clinic	
	type	event	
	event	move	
	agent	captain	
	theme	clinic	
	time	future	
	modal	[deontic	must]

Leaving from Miami,

Speech Act	implicit-confirm
Task	origin
Conversation Domain	communication

And, what city are you flying to?

Speech Act	request-info
Task	destination
Conversation Domain	task

(b) Communicator domain
Source: (Walker and Passonneau, 2001)

(a) SASO-ST domain

Figure 1.3: Examples of dialogue acts from different domains

and dialogue act. These methods need resources in the form of utterances annotated with dialogue acts as the training data. Another resource required by such an architecture is the information state update rules which update the dialogue state based on incoming & outgoing dialogue acts. E.g., An information-state based virtual human system used to train negotiation skills, SASO-EN (Traum et al., 2008b), contained 838 information state update rules. The NLU module used training data where around 4500 utterances were annotated with the corresponding dialogue acts. There were a total of 136 dialogue acts for NLU. The NLG module was trained with 75 dialogue acts linked to around 450 utterances.

Surface Text based architecture

Figure 1.4 represents an architecture where dialogue management happens at the surface text level. Here the input to the dialogue manager is surface text and the output is also surface text. Chatbots typically use such an architecture, where the response surface text is generated as a string transformation of the input text (e.g., Eliza (Weizenbaum, 1966) and AIML-based chatbots (Wallace, 2003b)). For such systems, the system designers need to author a set of pattern matching and string rewriting rules. A characteristic limitation of such rewriting approaches is producing syntactically incorrect or semantically invalid outputs.

Another approach commonly used for surface text based dialogue management is corpus-based retrieval (e.g., (Marinelli and Stevens, 1998; Chu-Carroll and Carpenter, 1999; Leuski et al., 2006)). This approach has an advantage of generating outputs which within themselves are valid in terms of syntax and semantics. There are also efforts that combine the string rewriting and corpus-based approaches by learning the pattern matching rules from a corpus (Abu Shawar and Atwell, 2003).

The resources required for such an architecture are either a set of string transformation rules or a corpus of stimulus-response. E.g., Alicebot uses a set of 41000 string transformation rules². SGT Star (Artstein et al., 2008) uses a corpus-based retrieval approach and has a corpus of around 300 responses and about 1850 stimuli linked to each other in a many-to-many relation with a total of around 2600 hand-authored links.

²Source: <http://www.alicebot.org/aiml.html>

Here we presented two architectures for dialogue systems based on the level at which the dialogue management takes place. But there are other criteria which can be used to distinguish various dialogue system architectures. Moreover, there are many possible variations based on the actual dialogue management techniques used. All these variations ultimately result in differences in resources required to build a dialogue system. E.g., One of the ways to avoid hand-authoring dialogue management rules is to learn them using reinforcement learning (Levin et al., 2000; Williams and Young, 2007; Heeman, 2007). These systems can learn what action to perform in a given dialogue state by running a large number of dialogue simulations using simulated users. For such systems, the system designers need to specify all possible actions, dialogue states and the reward structure. These specifications are best done at the dialogue act level which means utterances annotated with dialogue acts are one of the required resources.

Both the dialogue system architectures presented here (*dialogue act* based and *surface text* based), are suitable for virtual human dialogue systems. Each of them have their own set of required resources. There is a trade-off between the cost of building a specific set of resources and the performance of the resulting dialogue system. Generally, more resources lead to improved performance, but some of the resources required for building a dialogue system tend to be *application-dependent*. Collecting these resources is costly and time-consuming. In this thesis, we will identify and alleviate such problems towards the goal of rapid prototyping of a virtual human dialogue system. Next we look at the process of building a dialogue system and evaluating it.

1.3 Building a Dialogue System

Traditionally the development life cycle for a dialogue system is an iterative process and involves a series of steps as shown in Figure 1.5. Dialogue system designers start by defining the domain of interaction. This is followed by collecting human-human dialogue data in the form of roleplays and optionally corpus collection from Wizard-of-Oz (WoZ) experiments. For the dialogue act based architectures, the next step involves annotating the collected dialogue corpus with dialogue acts and associated semantics. The dialogue state is maintained by the dialogue manager and it requires either manually authoring the update rules or learning them by using reinforcement learning. The last process in the loop is to evaluate the dialogue system by conducting test runs with users and gathering evaluation metrics like user satisfaction, task completion, efficiency or appropriateness of responses.

During corpus collection, roleplays are generally collected first and the resulting dialogue corpus is used to validate and refine the domain of interaction. Once the domain of interaction gets relatively stable Wizard-of-Oz (WoZ) data collection can begin. In WoZ corpus collection a human (wizard) takes

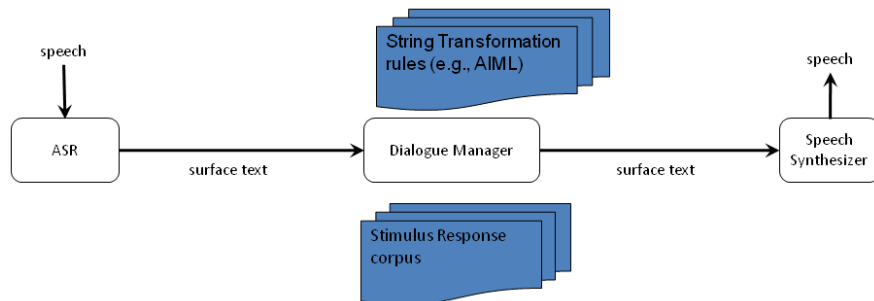


Figure 1.4: Surface Text based architecture of a spoken dialogue system and required resources

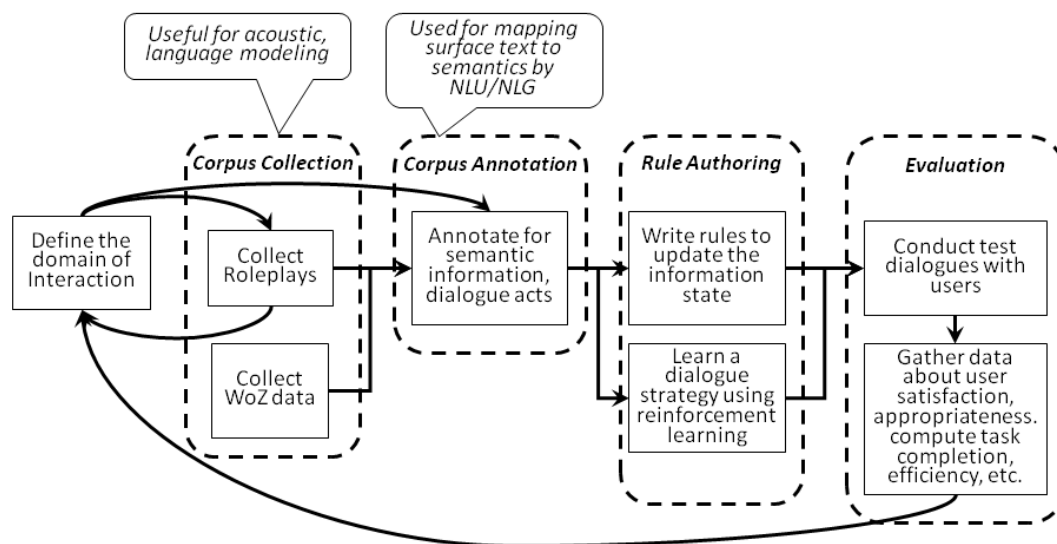


Figure 1.5: Process of building a dialogue system.

part in conversation on behalf of the system. The main difference between WoZ and roleplays is that the interaction with the wizard tends to be more restricted. E.g., Only pre-determined utterances/templates can be used as responses³. During corpus collection the dialogue data is transcribed and the resulting resources can be used for domain adaptation of acoustic and language models for speech recognition. This data also helps to fill in the gaps in the domain of conversation – task model, knowledge base, etc. Although this data collection activity is very useful, it is time-consuming and serves as a bottleneck in the rapid development of dialogue systems.

For dialogue systems that operate on the dialogue act level, the dialogue utterances are further annotated for dialogue acts and other semantic content (See Figure 1.3 for examples). These annotations can be used as training data for the NLU and NLG components. For rule-based systems like those employing information-state based dialogue modeling (Traum and Larsson, 2003), update rules are written that update the information state based on the incoming or outgoing dialogue act along with its semantic information. See Figure 1.6 for an example of such a rule. These processes of annotating the collected dialogue data and authoring the information state update rules serve as additional bottlenecks in the process of building a dialogue system.

Dialogue systems are evaluated by conducting dialogues with real or recruited users in as realistic situations as possible. These dialogues are then transcribed and annotated. Subjective and/or objective evaluation metrics are computed for these dialogue runs. Evaluation of a dialogue system is time-consuming, requires a lot of human involvement and hence is itself another bottleneck.

³For task-oriented speech interfaces, the difference between roleplays (human-human dialogues) and WoZ dialogues can be significant due to the application specific constraints on the interaction and the fact that users know that they are interacting with a computer. This is one of the main reasons for conducting WoZ studies in the first place (Dahlbäck et al., 1993). In case of virtual human dialogues, these differences should be less pronounced. We have observed the roleplay dialogues tend to have longer turns and more frequent speech overlaps compared to the WoZ dialogues (see Chapter 4 Section 4.1).

```

sp{top-state*apply*operator*update-dialogue-state*csa*assert
  (state <s> ^name          top-state          ^operator      <o> )
  (   <o> ^name            update-dialogue-state ^speech-input  <si> )
  (   <si> ^speaker        <speaker>            ^interpretation <i> )
  (   <i> ^conversation    <c>                  ^speech-act    <csa>)
  (   <c> ^grounding       <cgu>)
  (   <cgu> ^dialogue-history <csa>)
  (   <csa> ^action        assert                ^content      <sem>
    ^addressee          <addr>)
  -->
  (   <cgu> ^commitment    <comm> + &          ^conditional <cond> + & )
  (   <comm> ^type          commitment          ^holder     <speaker> )
  (   ^sanction            assertion            ^proposition <sem> )
  (   <cond> ^type          conditional          ^trigger     <accept> )
  (   ^consequent          <comm2>)
  (<accept> ^action        accept                ^content    <csa> )
  (   ^actor              <addr>                ^addressee  <speaker> )
  (   <comm2> ^type          commitment          ^holder     <addr> )
  (   ^committed-to       <speaker>            ^sanction    acceptance )
  ^proposition <sem>)
}

```

Figure 1.6: A sample information state update rule for SASO-ST virtual human dialogue system (Traum et al., 2005) written in SOAR 7.3 (<http://sitemaker.umich.edu/soar/home>). This domain-independent rule updates the information state upon receiving an *assert* dialogue act. The update reflects that the *speaker* is socially committed to the *proposition* of the dialogue act. It also adds a conditional effect that if the *addressee* accepts the assertion, the addressee is also socially committed to the proposition in the dialogue act.

1.4 Evaluating a Dialogue System

Dialogue systems need to be evaluated for various reasons. One might want to evaluate the system just to see to what degree the goals are being accomplished or to compare two or more systems with one another. Evaluation can also lead to understanding the shortcomings of the system and the reasons for these. Finally the evaluation results can be used as feedback in improving the system.

Generally, evaluating a dialogue system involves evaluating all of its components such as ASR, NLU, DM, NLG, Speech synthesizer. For this dissertation, we are mainly concerned with evaluating the dialogue modeling component. Evaluating a dialogue model requires making a series of decisions. Figure 1.7 shows a schematic representation of such decisions for evaluation of dialogue models for virtual humans.

The first decision is which evaluation metric to use. This is dependent on the goals of the dialogue system. In case of a task-oriented dialogue system, some suitable choices for an evaluation metric are user satisfaction, task success rate, task efficiency, etc. (Walker et al., 2000). For tutoring dialogue systems, some suitable evaluation metrics can be user satisfaction or learning gain as measured from differences between post-test and pre-test scores (Forbes-Riley and Litman, 2006). Since the goal for virtual humans is to be as human-like as possible, a suitable evaluation metric for virtual human dialogue systems is how appropriate or human-like the responses are for a given dialogue context.

The next decision is who evaluates the dialogue models. The dialogue models we need to evaluate are designed to be part of a virtual human who will engage human users in natural language conversations. Judging appropriateness of a response utterance given a dialogue context in such conversations is not an easy process and may require human-level intelligence. This is why human judges are a natural choice for evaluators.

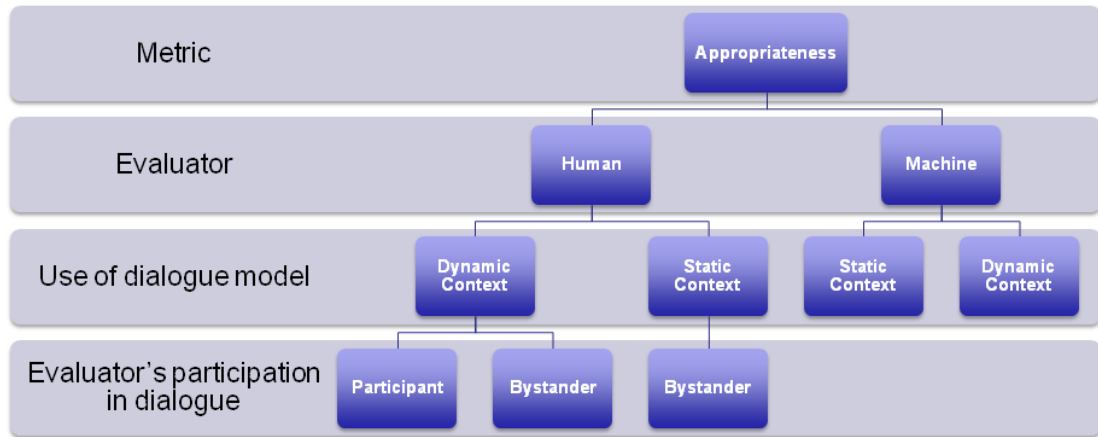


Figure 1.7: A schematic representation of various decision factors in evaluating dialogue models for virtual humans

Although humans are best suited to evaluate appropriateness of responses, using humans as judges is costly and time-consuming. For these and other reasons, automatic evaluation becomes an attractive alternative.

The next decision criterion is how the dialogue model to be evaluated is used in the process of generating response utterances and the corresponding dialogue contexts. There are two possible settings *Dynamic Context* and *Static Context*. Figure 1.8 shows a schematic representations for these different settings.

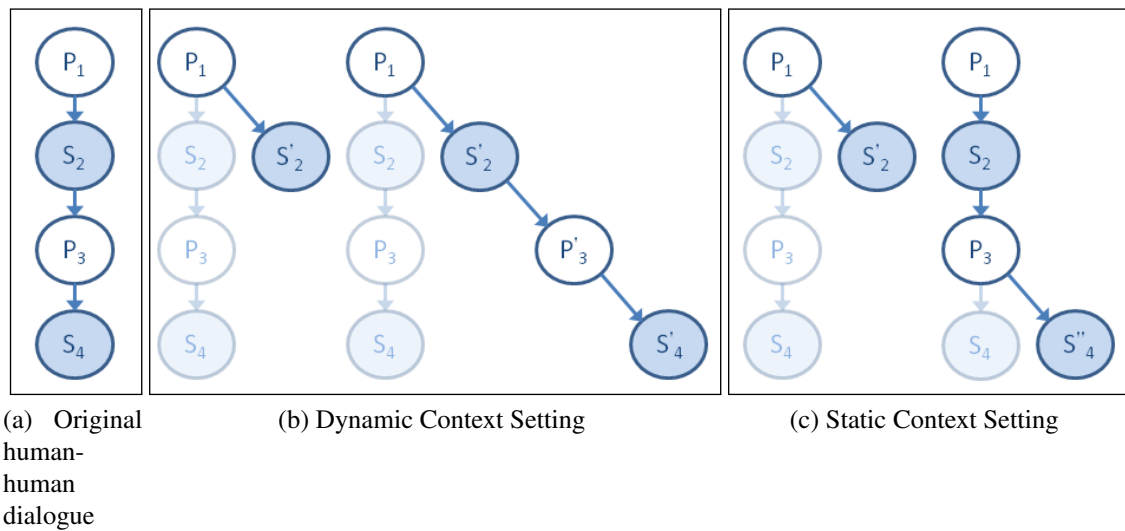


Figure 1.8: Schematic representation of *Dynamic Context* and *Static Context* evaluation settings.

Dynamic Context

In *dynamic context* evaluation, the dialogue model is used for generating the response utterances as well as the dialogue contexts with respect to which the subsequent responses are evaluated. In this case, we build a dialogue system using the dialogue model that needs to be evaluated. A human user interacts with this dialogue system. The system's response is the top-ranked response utterance for the given dialogue context as ranked by the dialogue model. This response then becomes part of the subsequent dialogue contexts.

Figure 1.8b shows the first two stages of the *dynamic context* evaluation process. At first, the user produce an utterance P_1 . Based on the context $\langle P_1 \rangle$, the dialogue model being evaluated produces the response utterance S'_2 . This response may be different from utterance S_2 , which was the response in the original human-human dialogue (Figure 1.8a). The user continues the dialogue and responds to the system's response with utterance P'_3 . The next response from the system produced by the dialogue model being evaluated is based on the context $\langle P_1, S'_2, P'_3 \rangle$. This context is dependent on the dialogue model being evaluated. Thus during dynamic context evaluation the resulting dialogue (and the intermediate dialogue contexts) are generated through an interactive process between a human user and a dialogue model. If an inappropriate response is chosen by the dialogue model then it becomes part of the context used to select the next response. Thus the dialogue model has the potential to recover from its errors or to build on them. System's responses are evaluated for appropriateness with respect to the same contexts that were used to generate them.

Static Context

In *static context* evaluation the dialogue model is used for generating only the response utterances. The dialogue contexts are not affected by the specific dialogue model being evaluated. These dialogue contexts are extracted from actual in-domain human-human dialogues. For every turn whose role is to be played by the system, we predict the most appropriate response in place of that turn given the dialogue context.

Figure 1.8c shows the first two stages of the *static context* evaluation process. The first system response is generated based on the context $\langle P_1 \rangle$ and is S'_2 , the same as in the case of *dynamic context*. But for the second response from the system, the context is reset to $\langle P_1, S_2, P_3 \rangle$ the same as the original human-human dialogue and does not depend on the dialogue model being evaluated. The system's response then is S''_4 , which can be different from both S_4 (human-human) and S'_4 (dynamic context). Again, the system's responses are evaluated for appropriateness with respect to the same contexts that were used to generate them.

The next decision criterion in evaluating dialogue models is whether the evaluator takes part in the conversation. If we require that the evaluator participates in the dialogue then each dialogue can be evaluated by only one evaluator – the participant himself. This evaluation scheme assumes that the conversational participant is in the best position to judge the appropriateness of the response. The Turing test (Turing, 1950) calls for such a dynamic context evaluation by the participant where instead of appropriateness, the evaluation metric is whether the conversational participant is human or machine.

Although evaluation by a dialogue participant is the most faithful evaluation possible, it is costly. As only one evaluator can judge a dialogue, we need to create a large enough test corpus by conducting conversations with the system. Moreover, volunteers may find playing two roles (dialogue participant and evaluator) difficult. In such cases, evaluation by a bystander (overhearer) can be a suitable alternative. In this type of evaluation the evaluator does not actively participate in the conversation and more than one evaluator can judge a dialogue for appropriateness of responses. In case of multiple judges, the average

of their judgments can be used as a final rating for appropriateness. For static context evaluation, the evaluator is always a bystander if s/he doesn't take part in creating the original human-human dialogue.

1.5 Bottlenecks in Dialogue System Development

The aim of this work is to identify the bottlenecks in rapid prototyping and evaluation of dialogue systems for virtual humans and to advance the state of the art by alleviating them. Now that the development life cycle for dialogue system has been examined, we can take a closer look at these bottlenecks. Most of the sources of the bottlenecks found in the process of building and evaluating a dialogue system are *application dependent*. Resources created as a result of developing one application cannot be easily used in another application and the process has to start all over again for a new application.

Corpus Collection

Corpus collection, in the form of roleplays or WoZ exercises, must be performed within the domain of a specific application. This data is useful for refining the domain of conversation (task models and knowledge bases). It is also used to train acoustic models and language models for speech recognition. If the dialogue system is using the surface text based corpus retrieval approach then corpus collection is a prerequisite. If a dialogue system acts at the dialogue act level, then one of the reasons for collecting a large corpus is to ensure the *completeness*. Here completeness implies that all the dialogue acts that can be reasoned with by the dialogue manager should be seen at least once. If the corpus collection task is to be bypassed we need to take care of creating the resources for ASR, NLU and NLG that ensure completeness within the specific application domain.

Corpus Annotation

For the dialogue act based architecture, we need NLU and NLG components to map surface text into an abstract representation (dialogue acts) and back. If these components employ statistical techniques, then the training data needs to be in the form of utterances annotated with dialogue acts. Most of the dialogue act taxonomies and ontologies for the corresponding propositional content are application specific (See figure 1.3 for examples of application-specific dialogue acts).

There have been a few proposals for standardizing dialogue act taxonomies. The DAMSL scheme (Core and Allen, 1997) allows the utterances to have dialogue acts annotated on several layers (forward-looking, backward-looking, information level, etc.). The DIT++ scheme (Bunt, 2006) accounts for multiple functions of utterances and allows annotations on multiple dimensions. Recently, an ISO standard for dialogue act annotation has been designed (Bunt et al., 2012). Despite these efforts, many system developers find such dialogue act schemes too general to be useful, especially for generation (Alexandersson, 1996). Verbmobil uses a custom set of dialogues acts specifically designed for appointment scheduling (Jekat et al., 1995). Jurafsky et al. (1997) follow the DAMSL annotation scheme but with certain modifications to best predict the next dialogue act. These modifications include splitting the statement category into statement-not-opinion and statement-opinion. Traum (2000) provides a list of questions that can influence the choice of dialogue act taxonomy.

Rule Authoring

For information state based dialogue management, information state update rules need to be written. These rules can be categorized as domain-independent or domain-dependent (For examples,

see Figure 1.6 and 1.9 respectively). One of the primary responsibilities for the dialogue manager is to come up with a response. This may involve interacting with the task model or external databases which are application specific. This leads to a substantial amount of application-dependent rules. Moreover the rules are specific to the dialogue act set being used which itself maybe application specific.

```
sp{top-state*propose*operator*output-speech*doctor*threatened-by-attack
  (state <s> ^name top-state ^agent-name <me>
    ^conversation <c> ^things-i-said <ts>
    ^social-state <ss> )
  ( <ss> ^commitment <comm> )
  ( <c> ^active-participant <me> ^active-participant <hearer> )
  ( <comm> ^holder <hearer> ^committed-to <me>
    ^proposition.reference <ref> )
  ( <ref> ^name attack-by-americans )
  -( <ts> ^said-string dwob-string26 )
-->
  ( <s> ^operator <o> + =)
  ( <o> ^name output-speech ^priority-class respond
    ^conversation <c> ^goal <b> )
  ( <b> ^action accuse ^type backward
    ^addressee <hearer> ^reason <comm>
    ^speaker <me> )
}
```

Figure 1.9: A sample information state update rule for SASO-ST virtual human dialogue system (Traum et al., 2005) written in SOAR 7.3 (<http://sitemaker.umich.edu/soar/home>). This is a domain-dependent update rule which proposes a response when the virtual human agent believes that the other participant (here, trainee captain) is socially committed to “attack-by-americans”. The proposed response is reply back with an accusation, “you are the threat i need protection from you”.

For surface text based architectures, such as AIML based chatbots, system developers need to author string transformation rules that can match the surface text phrases or keywords. These phrases or keywords are specific to the language being used and will always be application dependent.

Evaluation

Evaluation of a dialogue system depends on its goals. In order to evaluate a dialogue system, test runs need to be conducted with that dialogue system. Collecting this test corpus is a required step and is application-specific. For task oriented dialogue systems, the resulting data needs to be annotated to compute metrics like task completion rate, task efficiency, etc. Another metric commonly used is user satisfaction – a subjective measure, which requires users to fill out the questionnaires. Non-task oriented systems like virtual humans are evaluated with utterance by utterance appropriateness ratings. But all these methods have substantial human involvement. If the dialogue management is based on machine learning methods, then frequent evaluations are required which increases the amount of effort required.

Expertise Required

Corpus annotation, rule authoring and evaluation all involve substantial human effort and expertise. The corpus annotation task requires expertise for designing a dialogue act scheme and annotating utterances with it. This involves understanding of dialogue acts and associated semantics. Figure 1.3 illustrates the complexity of dialogue act annotation task. Rule authoring for information state based dialogue management involves understanding of rule based systems and theories of dialogue such as issues under negotiation (Larsson, 2002) or social commitments and obligations

(Poesio and Traum, 1998). The example rule from Figure 1.6 is based on this Poesio-Traum theory. It is very difficult to avoid both the bottlenecks of corpus annotation and rule authoring while building dialogue systems. This makes required expertise by developers a bottleneck.

1.6 Trade-offs in Dialogue System Development

We’ve presented two types of architectures – dialogue act based and surface text based. Both of these are suitable for building a virtual human dialogue system. We’ve also presented the bottlenecks associated with building a dialogue system. Not every dialogue system will encounter all the above bottlenecks. In the dialogue act based architecture, a system that uses a hand-written rule-based parser for NLU can do without corpus annotations. A system that use reinforcement learning to learn a strategy won’t require explicit manual rule authoring. Dialogue systems using strict directed dialogue can avoid corpus collection. But not all the bottlenecks can be avoided. The cost of accumulating the required resources such as dialogue corpus, dialogue act annotations and update rules is high. The benefits associated with such a high cost are ability to model complex behaviors, detailed explanation of dialogue behaviors, etc. In surface text based architectures, specifically the corpus retrieval approach, the bottlenecks of corpus annotation and rule authoring are absent. There are relatively fewer requirements on resources. Only an un-annotated dialogue corpus needs to be collected but complex dialogue behaviors and explainability cannot be modeled.

The specific architecture chosen is dictated by the specific goals of the dialogue system and its evaluation criteria. This architecture, in turn, determines the specific types of resources required. There is a trade-off between the cost of building a specific set of resources and the performance of the resulting dialogue system. Generally, more resources lead to improved performance, but some of the resources required for building a dialogue system tend to be application-dependent. Collecting these resources is costly and time-consuming. Currently there is no way to make an informed decision regarding this trade-off.

In order to gather useful information regarding these trade-offs, two sub-problems must be addressed. First, as most of the traditional dialogue architectures impose their own constraints on which resources can be combined, some resource combinations may not have been explored. We need flexible architectures that allow exploration of such resource combinations. Second, evaluating the performance of the resulting dialogue system is a time-consuming and costly process, as it generally involves human participation. We need the ability to easily predict the performance of a dialogue system, so as to evaluate a wide variety of resource combinations than is currently feasible.

1.7 Thesis Statement

The goal of this thesis is to enable cost-effective and rapid prototyping and evaluation of dialogue systems. Our work addresses both types of architectures used for virtual human dialogue systems.

- In the case of dialogue-act based architecture, once the type of resources to be used have been determined, the cost of developing a dialogue system can be lowered by reducing the cost of building those specific resources. Our approach to reducing this cost is to allow *non-experts* to build such resources with the help of *integrated authoring tools*.
- For surface-text based architectures, we posit that for certain virtual human conversational domains, a response to a dialogue context can be formulated by simply selecting an appropriate response from a set of utterances from an in-domain human-human dialogue corpus.

- We develop flexible architectures that allow novel combinations of different types of resources, such as surface text transcripts and information state annotations. This allows us to better understand the relative utility of different resources and the resulting information can be used to optimize the cost and/or performance of the dialogue system.
- In order to explore a wide variety of resource combinations, we need to reduce the cost of evaluating the performance of a dialogue system. We develop automatic evaluation measures, which do not require human participation, and correlate well with human judgments.

1.8 Thesis Contributions

The contributions of this thesis are to identify the bottlenecks involved in rapid prototyping of dialogue systems and to advance the state of the art by mitigating these challenges. Depending on the requirements of the dialogue system and its architecture, we can choose to incorporate the following solutions to alleviate the bottlenecks. The work described in this thesis is done within the scope of *Virtual Human Dialogue Systems*. The specific contributions of this thesis towards the practical goal of rapid prototyping and evaluation of virtual human dialogue systems are as follows:

- **Reducing the cost of collecting specific resources with integrated authoring tools** (*chapter 3*)
Addresses bottlenecks: *Corpus Collection, Corpus Annotation, Expertise Required*
Architecture: *Dialogue Act based*

Dialogue Act based architectures allow for deep understanding of dialogue progression and can model arbitrarily complex behaviors. But this comes at a cost of annotating a dialogue corpus with dialogue acts and authoring information-state update rules. Collecting these resources is costly and time-consuming. Several toolkits and authoring environments have been proposed for reducing the authoring burden. But some of these toolkits do not lessen the expertise needed, while others don't allow sophisticated dialogues. E.g., toolkits such as RavenClaw (Bohus and Rudnicky, 2003), TrindiKit (Larsson et al., 2004), Midiki (MITRE, 2005) which have been used for developing sophisticated task-oriented dialogue systems require considerable expertise in design of dialogue systems, theories of dialogue and software development. There have been toolkits designed for *non-expert* users such as the CSLU toolkit (Sutton et al., 1998), which allowed *non-experts* to build finite-state based dialogue systems allowing only simple directed dialogues and NPCEditor (Leuski et al., 2006), a tool that allows *non-experts* to author simple question-answering virtual humans where resulting dialogue systems are unable to track coherence relationships over utterances other than immediately adjacent ones. Currently, there is no existing authoring approach where non-experts can develop Advanced Question-Answering Virtual Humans.

The first contribution of the thesis is an authoring process designed for *dialogue act based* architectures and the genre of Advanced Question-Answering dialogues which allows *non-experts* to author the required resources rapidly. We demonstrate its viability by implementing the necessary integrated authoring tool and having *non-experts* build Advanced Question-Answering Virtual Humans (Gandhe et al., 2008). Our authoring process and the accompanying tool allows *non-experts* to build systems faster (within a few weeks) compared to what experts used to be able to do without the tool (up to several months).

There are 4 types of resources required for such a dialogue system: domain of interaction, a set of dialogue acts, surface text examples associated with the dialogue acts and a set of information-state update rules. Our authoring approach allows dialogue system development to begin without

the initial corpus collection and follows a top-down approach starting from authoring the domain knowledge for the virtual characters. The novelty of our approach is that the authoring tool automatically generates required dialogue acts based on the specified domain and ensures *completeness* and *consistency*. Here *consistency* refers to generating only valid dialogue acts that can be correctly handled by the dialogue manager and *completeness* refers to generating all dialogue acts that are relevant with respect to the character’s domain knowledge. Currently the tool supports 39 dialogue act types and 13 types of dialogue behaviors (e.g., question-answering, elicit-offer-response, grounding, greetings, closings, etc.) which are controlled by expert-authored genre-specific but domain-independent information state update rules.

To date, 10 advanced question-answering virtual humans have been developed by *non-experts* and these characters perform at par with the expert authored characters, but require considerably lower development time, with some characters being developed in as low as two weeks (Gandhe et al., 2011). The resulting virtual human characters have been used for training tactical questioning skills, negotiation skills (Artstein et al., 2009b) and for acting as confederates in psychology experiments (Khooshabeh et al., 2011). We have also evaluated the genre-specific dialogue act scheme and have found that around 76% of the user’s utterances are covered by the dialogue acts supported by the tool (Artstein et al., 2011).

- **Viability of the Selection approach** (*chapter 4*)

Addresses bottlenecks: Rule Authoring, Expertise Required, Corpus Annotation

Architecture: Surface Text based

There are two approaches for formulating a response in surface text based dialogue systems: *Generation* and *Selection*. In the generation approach, the response utterance is dynamically constructed using some composition procedure (e.g., grammar rules) from semantic units or surface text phrases. Authoring such grammar rules requires expertise and learning such rules automatically requires a dialogue-act annotated corpus. In the selection approach, the response is simply selected from a set of pre-existing utterances, generally collected from a dialogue corpus. The selection approach has an upper limit based on whether an appropriate response exists in the set of pre-existing responses. It is difficult to know which of these two approaches is most suited for a given dialogue domain. **In this thesis, for the first time in literature, we propose an empirical method to determine whether the selection approach is viable for a given domain and apply it to 10 different domains.** The theoretical maximum performance of the selection approach is the proportion of test utterances for which an exact or approximate match exists in the corresponding training corpus. For virtual human applications such as question-answering and negotiation where virtual humans are situated within the context of a narrative, we find that the selection approach is suitable. In such domains, over half of the test utterances are seen before in the corpus (Gandhe and Traum, 2010).

- **Developing flexible dialogue architectures that allow novel combinations of different types of resources** (*chapter 4*)

Addresses bottlenecks: Corpus Annotation, Rule Authoring, Expertise Required

Architecture: Flexible Architecture (Starting from Surface Text based)

Surface text based architectures require relatively low-cost resources such as dialogue transcripts. But without high-level information state representations such dialogue systems cannot adequately model complex behaviors. There have been efforts in combining resources, such as surface text from dialogue transcripts and dialogue act annotations, with a goal of predicting the next dialogue

act (e.g., (Stolcke et al., 2000)) or learning the structure of the dialogue (e.g., (Bangalore et al., 2006b)). Chotimongkol and Rudnicky (2008) have tried to use an unannotated dialogue corpus to learn the structure of the domain of the dialogue system. But these efforts fall short of actually implementing and evaluating a dialogue system. Lee et al. (2006) have used example based dialogue management for a limited domain task-oriented system but the example dialogues need to be annotated with dialogue act, user intention and dialogue history.

There is a trade-off between the cost of building a specific set of resources and the performance of the resulting dialogue system. To better understand this trade-off there is a need for flexible dialogue system architectures that allow novel combinations of resources. Moreover, evaluating dialogue systems that operate at surface text level is not trivial. **In this thesis, we develop flexible architectures that allow novel combinations of different types of resources, such as surface text transcripts and information state annotations, and systematically evaluate them in three different evaluation settings.**

For the first time we explicitly model these choices involved in the evaluation process (See Figure 1.7). We have evaluated the dialogue models in different settings – *dynamic context* or *static context*, and with different evaluators – dialogue participants themselves or bystanders, and with different modalities – typed text or speech.

In order to understand the relative utility of different types of resources and the effect of evaluation setting, we conducted a series of experiments. We have implemented eight unsupervised dialogue models that work primarily at the surface text level and use different combinations of resources. These models can be bootstrapped from un-annotated in-domain human-human corpus. Resources such as information state annotations for topics can be incrementally added to the dialogue model as they become available. We have evaluated these dialogue models by eliciting human judgments for appropriateness of response utterances given the dialogue context. We have compared these models with lower random baselines and upper human-level baselines.

In dynamic context evaluation by dialogue participant themselves, the *Text Segmented Nearest Context* model which uses the information state annotations with topic signatures in addition to the surface text of the context performs significantly better than the model *Text Nearest Context* which uses the context information alone (Gandhe and Traum, 2007b). In dynamic context evaluation by bystanders, the best performing model *Text Segmented Nearest Context* can achieve up to 68% of human-level performance above the *Text Random* baseline (Gandhe and Traum, 2007a). This baseline chooses an utterance from the scenario-specific corpus at random. In static context evaluation, the advanced text-to-text models which use the utterance content information in addition to context information perform better than the simpler *Nearest Context* model which uses context information alone.

As far as we know this is the first time *Perceptron* models have been used for surface-text based dialogue modeling. *Perceptron* models are suitable candidates for flexible architectures as they allow novel combinations of resources by simply changing the set of features used (Gandhe and Traum, 2013). The systematic evaluations we performed for different dialogue models also form the basis for the next contribution towards automatic evaluation for dialogue models.

- **Reducing the cost of estimating the performance of a dialogue system (chapter 5)**
Addresses bottlenecks: Evaluation, Expertise Required
Architecture: Surface Text based

For *Surface text based* architectures, evaluating the performance of the resulting dialogue system involves collecting subjective judgments about the appropriateness of responses given the dialogue

context. Since the evaluation process requires a lot of human participation, it is time-consuming, costly and forms another bottleneck. Moreover for dialogue models that employ machine learning, we need to repeatedly evaluate the performance of the dialogue system with different possible parameter settings. Our approach towards reducing the cost of estimating the performance of the dialogue system is to reduce the human involvement in evaluation. **As a final contribution, we have evaluated two previously proposed automatic evaluation measures in terms of how well they correlate with human judgments and have developed two new measures, that achieve higher correlation with human judgments.**

Recently, DeVault et al. (2011b) have proposed an automatic evaluation method, *Weak Agreement*, for evaluating dialogue models in static context setting, but they do not evaluate the metric for how closely it correlates with human judgments. In this thesis, we have evaluated *Weak Agreement* for how well it correlates with human judgments (Pearson’s $r = 0.80$) and have proposed and evaluated another evaluation metric, *Voted Appropriateness*. The Voted Appropriateness metric is based on the data collected from wizards and correlates very well with human judgments (Pearson’s $r = 0.89$). In this thesis, we have also used the *Weak Agreement* automatic evaluation as feedback for manually tuning free parameters for the *Cross-lingual Relevance Model* and *Perceptron* models we proposed earlier.

For dynamic context setting, Lapata (2003) had proposed using information ordering for evaluating discourse coherence models and had proposed Kendall’s τ as an automatic evaluation metric. Ritter et al. (2010) have applied this information ordering task and used Kendall’s τ for evaluating dialogue models. But they did not verify whether the same automatic metric works well for the dialogue setting. We have evaluated the previously proposed Kendall’s τ for information ordering task as applied to dialogues and found it has a weak correlation with human judgments (Pearson’s $r = 0.33$). In this thesis, we also proposed an automatic evaluation metric, $(b_2 + b_3)/2$, which is based on the fraction of local orderings preserved during the information ordering task. Our metric achieves improved correlation with human judgments (Pearson’s $r = 0.75$) (Gandhe and Traum, 2008). These objective evaluation measures can be used to compare different dialogue models using different resources.

1.9 Outline

The rest of the dissertation is organized as follows. Chapter 2 presents an overview of related work in dialogue modeling.

Chapter 3 presents a practical way for rapid and cost-effective prototyping of virtual human dialogue systems. First, we elaborate on the *genre* of advanced question answering in section 3.1. This genre dictates the use of a dialogue act based architecture as described in section 3.2. *Non-experts* can build resources required for a dialogue system with the help of an integrated authoring tool. As described in section 3.3, the authoring process begins in a top-down fashion with the specification of the domain of interaction. The integrated authoring tool, DomainEditor presented in section 3.4, allows authors to edit the domain and also the surface text corresponding to all the automatically generated dialogue acts. Section 3.5 describes the dialogue manager used in conjunction with the authoring tool. It is based on information-state update mechanism (Traum and Larsson, 2003) and uses conversational game theory (Lewin, 2000) to track individual subdialogues. We evaluated this approach by building dialogue systems for new virtual human characters that take part in advanced question answering as described in section 3.6.

Chapter 4 presents flexible dialogue architectures that allow novel combinations of resources. These dialogue models work primarily at the surface text level and do not require corpus annotation or rule

authoring. In section 4.1, we describe the testbed application – SASO-ST, a virtual human engaged in a negotiation dialogue. Next in section 4.2 we present the surface text based dialogue system architecture and our simplifying assumptions. Section 4.3 contrasts two different approaches towards formulating a response. Our unsupervised dialogue models employ the selection approach and the viability of the selection approach is discussed in section 4.4. Section 4.5 discusses different possible selection criteria that could be used with the selection approach and provides justification for the ranking approach we use. In section 4.6 we present six dialogue models including two baseline models. These models capture different aspects of local and global context of the dialogue and allow for novel combinations of resources. We have evaluated these models in *dynamic context* setting (section 4.7.1) as well as *static context* setting (section 4.7.2). We compare these models with an upper baseline of human performance and also the effect of input modality – speech or text. In section 4.8, we discuss some issues arising in unsupervised dialogue models such as annotating topic signatures at surface text level, granularity of utterances, etc. At the end, we list possible applications for the dialogue models in section 4.9.

Chapter 5 discusses the progress made in search for suitable evaluation understudy measures to evaluate the dialogue models. We investigate both the settings – *static context* and *dynamic context*. In section 5.1, we investigate automatic evaluation for the static context setting. We examine a previously proposed, but not evaluated evaluation measure, *weak agreement*, for how closely it matches with human judgments in section 5.1.1. In section 5.1.2, we suggest an improved evaluation understudy, *voted appropriateness* and correlate it with human judgments for appropriateness. For dynamic context setting we propose to use the information ordering task as described in section 5.2.1. Section 5.2.2 describes various objective metrics that are used to judge the success of this information ordering task. We set up experiments to validate whether this task can be used to evaluate the dialogue coherence models (section 5.2.4) and to evaluate how well these objective metrics correlate with human judgments (section 5.2.5).

Finally we conclude with chapter 6. We summarize the results from this dissertation and present avenues for future work.

Chapter 2

Related Work

This chapter summarizes the background work relevant to the thesis contributions. Different architectures have been proposed for building dialogue systems. Each architecture in turn requires a specific set of resources. In order to achieve the goal of rapid prototyping and evaluation of dialogue systems, we need to reduce the cost and time required for collecting these resources. These resources can either be collected manually or can be learned from dialogue corpora. The first contribution of the thesis is towards reducing the effort required for manual resource creation. In section 2.1 we present an overview of previous work related to authoring tools. The second and third contributions of the thesis are about developing unsupervised dialogue models from dialogue corpora. Section 2.2 presents the related work employing data-driven methods for learning useful dialogue resources. Here, we organize the discussion based on different types of information that can be learned from dialogue corpus such as, dialogue structure, domain structure, dialogue strategy, dialogue coherence models, etc. The final contribution of the thesis is about reducing the cost of evaluating dialogue systems and in section 2.3, we summarize the related work in evaluating dialogue systems.

2.1 Authoring Tools for Dialogue Systems

Many toolkits and authoring environments have been developed for aiding dialogue system resource creation. Some of these also aim at lowering the required expertise. The Rapid Application Developer from CSLU toolkit (Sutton et al., 1998) allowed designers to build dialogue systems employing finite state dialogue models. The authoring environment was accessible by non-experts and allowed building systems that could conduct simple directed dialogues. These are *system-initiative* dialogues where the flow of the conversation is being controlled by the system. The advanced question answering systems we are interested in engage the trainee in a natural conversation. The input from the interviewer is more free-form than the input for directed dialogues. Advanced question answering dialogues are mostly *user-initiative* where interviewers are asking most of the questions. Our dialogue system also allows for some initiative from the virtual character required to engage in simple negotiations (e.g., character eliciting different offers). This mixed-initiative behavior is more challenging than simple system directed dialogue.

There have been several commercial dialogue building solutions based on VoiceXML (McGlashan et al., 2004), which allows for a simple form-based dialogue management (Goddeau et al., 1996). There are dialogue system frameworks that allow more complex dialogues. RavenClaw (Bohus and Rudnicky, 2003) is another dialogue architecture where designers can specify hierarchical domain task specification. The dialogue management in RavenClaw builds on top of agenda based dialogue management

technique (Xu and Rudnicky, 2000). Although this architecture has been successfully used for building multiple dialogue systems, it is most suited for task-oriented dialogue systems and using it requires considerable expertise in software development and design of dialogue systems. Other dialogue system architectures such as TrindiKit (Larsson et al., 2004) or Midiki (MITRE, 2005), which use information state based dialogue modeling (Traum and Larsson, 2003) have the same issue. These systems require considerable knowledge of the dialogue theories and software development. For conversational dialogue systems, standards like AIML (Wallace, 2003a) have been defined. AIML rules specify how a stimulus is converted into a response using pattern matching and string transformations. Although writing AIML rules is relatively easy, maintaining systems with a large number of rules can be complex. Also sometimes inappropriate pattern matching leads to un-grammatical responses.

There have been some efforts in the area of tutorial dialogue systems that concentrate on building authoring tools which can be used by *non-experts* for rapidly building a dialogue system. TuTalk (Jordan et al., 2007) is one such system. The TuTalk authoring tool allows tutorial system researchers who may not have expertise in the dialogue system design to rapidly prototype dialogue systems and experiment with different ideas. The TuTalk authoring tool allows authoring of initiation-response pairs along with many features suitable for tutorial dialogue systems.

The NPCEditor (Leuski et al., 2006; Leuski and Traum, 2011) authoring tool allows *non-experts* to author simple question-answering virtual humans. Dialogue system designers author a set of questions, a set of answers, a set of links between questions and answers whenever the answer can be an appropriate response to the question. In this architecture, the resulting dialogue system is unable to track coherence relationships over utterances other than immediately adjacent ones. There was previously no authoring approach where non-experts could develop Advanced Question-Answering Virtual Humans. Our approach is to design an authoring tool that is specialized for a specific genre of dialogue viz. advanced question answering specifically tactical questioning and is designed for use by *non-experts*.

2.2 Data-driven approaches for Dialogue Modeling

Another contribution of this thesis is towards unsupervised dialogue modeling. Traditional dialogue management methods such as form-filling (Goddeau et al., 1996), agenda-based (Xu and Rudnicky, 2000) and information-state based (Larsson and Traum, 2000; Traum and Larsson, 2003) operate at the dialogue act level and require the dialogue system designer to specify the form structure in detail or author the information state update rules. These methods also require a separate NLU component, which maps surface text utterances to dialogue acts and may itself need a training corpus in the form of utterances annotated with dialogue acts. One of the ways of reducing the authoring burden is to use authoring tools as seen in section 2.1. Another way of alleviating the problem of resource creation is to acquire these from other more readily available resources such as dialogue transcripts. There has been work in data-driven approaches for learning dialogue structure, learning structure of the domain of interaction, learning dialogue strategies and formulating the system response utterance.

Learning Dialogue Structure

There have been some efforts at learning dialogue structure from a dialogue corpus. The learned structure is some sort of sequence model for dialogue acts. These dialogue acts may be simply illocutionary force types or partially specified by semantics related to the dialogue domain. Woszczyna and Waibel (1994) used hidden markov models (HMM) to infer the underlying dialogue structure for a spontaneous scheduling task in the context of a speech to speech translation project. Kita et al. (1996) tried to learn probabilistic dialogue structure from a sequence of illocutionary force type (IFT) labels annotated on

conference secretary dialogues. There has been a lot of work in dialogue act prediction and recognition: language model based classifiers (Reithinger and Klesen, 1997), transformation based learning (Samuel et al., 1998), HMM based models (Chu-carroll, 1998; Stolcke et al., 2000). The goal is to predict the next dialogue act given a dialogue history using features extracted from the sequence of previous utterances (words, prosody, dialogue act) etc.

Some researchers modeled dialogue using a hierarchical structure. Alexandersson (1996) modeled appointment scheduling dialogues with a 4-level hierarchical intentional structure. Bangalore et al. (2006a,b) modeled customer-agent catalog dialogues in a similar hierarchical structure. Both these efforts used a dialogue act annotated corpus and used an application specific dialogue act scheme.

The learned dialogue structure can be used to predict the next dialogue act and/or subtask labels. This information can be used towards improving the speech recognition accuracy (Stolcke et al., 2000), but none of these approaches have been used for response formulation.

Learning Domain Structure

Another line of research is to acquire the structure of the underlying domain from a human-human dialogue corpus. This is different from the recognition of some pre-defined structure of the dialogue.

Siu and Meng (1999) proposed a semi-automatic method for acquiring domain-specific semantic knowledge from an unannotated corpus. Working within the ATIS domain, they used agglomerative clustering for inducing semantic classes (e.g., city_names, airline_names) and temporal clustering to create phrasal structures. Chotimongkol and Rudnicky (2002) look at a similar problem that of automatically identifying concepts and the members (words) that belong to that concept within the domain of CMU Communicator. Chotimongkol and Rudnicky (2008) provide a method to completely identify the domain-specific dialogue information from human-human dialogue corpus. They model dialogue using a form based structure where the form is a set of concepts that can take some values. Their work showed that the corresponding form structure can be learned for task-oriented dialogues such as the air travel domain and maptask (Carletta et al., 1997) dialogues. The approach leaves room for the experts to intervene and improve the final learned form structure. The claim is that this method will cut down on the human effort involved in building dialogue systems but no explicit evaluation for this effort reduction is performed. Also this process has not yet been used to build a dialogue system.

Learning Dialogue Strategy (Reinforcement Learning)

The core of the dialogue management problem (i.e., formulating the response utterance given the dialogue context) can be viewed as a sequential decision making process. Reinforcement learning methods can be used for learning a dialogue strategy that updates the current state of the dialogue based on the observation of the incoming user utterance and formulates the response utterance/dialogue act based on the updated state. This type of learning addresses the bottleneck of writing information state update rules. A Markov Decision Process (MDP) can be used to represent the problem of optimizing such sequential decisions (Levin et al., 2000). Such an MDP can be specified by defining a set of states the dialogue system can be in, a set of actions that can be performed in those states (e.g., asking questions to the user, displaying the results to the user, performing database queries etc.) and a reward function which specifies how useful is the action given the current state. A dialogue strategy is learned so as to optimize the reward accumulated by the dialogue system. Generally for dialogue systems this reward function encodes the dialogue designer's intuitions about what it means to have a successful dialogue. The reward function can be modeled as a linear combination of various dialogue features such as dialogue length, number of results presented back to the user, task completion, etc. Designing the reward function is a very tricky task that often involves a lot of manual tuning (Paek and Pieraccini, 2008). Rieser and Lemon

(2008) have used the PARADISE evaluation framework (Walker et al., 2000), which predicts user satisfaction as a linear combination of the dialogue features, to define the reward function. Another important component of an MDP is the transition probabilities, which predicts the next state given the dialogue manager performed a specific action in a specific state. This encodes the uncertainty involved in user's responses to the system prompts or external events such as database lookups, etc. These transition probabilities are not readily available, instead a simulated user is used for finding the optimal policy through reinforcement learning. The user simulation module predicts the user's response based on the current state of the dialogue. Generally the state description for predicting the user's response is not as detailed as the state description used to come up with a system's response. In fact, a cooperative user can be easily modeled by conditioning his/her response only on the system's immediate prompt (Levin et al., 2000). The stochastic model for a simulated user can be learned from annotated corpora of dialogues in the same domain. Recently the same models used for building a dialogue system like agenda-based models have been used in user simulation (Schatzmann et al., 2007). Building a simulated user is as complicated as building the dialogue system in the first place. A more advanced variant, Partially Observable Markov Decision Process (POMDP), can be used handle the uncertainty due to speech recognition errors and hidden user intentions (Williams and Young, 2007).

Reinforcement learning based approaches can benefit from human-human dialogue data by either using it to model user simulations or by bootstrapping an initial policy for the dialogue system. Williams and Young (2003) use Wizard-of-Oz data to bootstrap such a supervised policy for an MDP-based dialogue system. Rieser and Lemon (2006) use a similar approach for learning a supervised policy for multimodal clarification dialogues. The reinforcement learning approach solves the problem of writing information state update rules at the cost of manual effort required in careful state space planning and tuning of the reward function. Also it requires dialogue-act annotation work since the states and actions are defined in terms of dialogue acts.

Example Based Dialogue Systems

Example based dialogue systems can generate a response to a dialogue context based on observed example dialogues. These example dialogues can be used to create a list of ⟨dialogue context, response⟩. The dialogue context is represented by various features such as user utterance, dialogue act, main domain action, component slots, etc. Lee et al. (2006, 2008) implemented such a system for a simple domain of electronic program guide (TV-guide). In cases where there is no good matches, the system falls back on manually written rules for dialogue flow. For their system the dialogue corpus needs to be tagged with the necessary information state features. If the dialogue context is as simple as the incoming user utterance, then the observed example data can be easily increased using methods like automatic paraphrasing. Using the method of Variant Transduction (Alshawhi and Douglas, 2001) a call routing dialogue systems was built starting with seed examples of utterance-action pairs with as little as 10 utterances per action.

Our solutions for unsupervised dialogue models, specifically models described in section 4.6.2–4.6.4, are very similar to these example based dialogue models. Our dialogue models operate in the domain of virtual human negotiations which are more complex than simple command & control or call routing applications. Also our dialogue models do not require any annotations and can be bootstrapped from dialogue transcripts.

Discourse & Dialogue Coherence Models

There has been substantial amount of work in discourse coherence modeling with applications to summarization, generation and automatic scoring of essays etc. Barzilay and Lapata (2005) used an entity grid model based on centering theory (Grosz et al., 1995). Certain entities that are salient in one part

of the discourse tend to occur again and again in prominent roles. The entity grid learns local transition patterns of how these entities change roles (e.g., subject, object, other, not mentioned). The models can explain the noun phrases in the sentences but not all words (e.g., verbs which signal intentions). [Soricut and Marcu \(2006\)](#) built a mixture model for discourse coherence based on the entity-grid model ([Barzilay and Lapata, 2005](#)), HMM model ([Barzilay and Lee, 2004](#)) and IBM model-1 ([Brown et al., 1993](#)). These models are evaluated using the information ordering task, where information-bearing elements (e.g., sentences) are ordered in the most coherent way and this ordering is either manually evaluated for coherence or compared to original human authored coherent ordering.

Following ([Barzilay and Lapata, 2005](#)), [Purandare and Litman \(2008\)](#) looked at the problem of modeling dialogue coherence in terms of information ordering. They developed a binary classifier model which can distinguish coherent dialogues from incoherent ones. The coherent dialogues were real human-human dialogues from switchboard corpus ([Godfrey and Holliman, 1993](#)) and incoherent ones were random permutations of the original ones. Their model used local transition patterns of lexical and semantic features. Their experiments only evaluated the performance on this binary classification task of coherent Vs incoherent. It is not entirely clear how this framework can be used in evaluation or simulation of dialogues.

[Leuski et al. \(2006\)](#) trained a Cross-lingual relevance model ([Lavrenko et al., 2002](#)) from a list of Question-Answer links. The model predicts what words should be in the answer given an incoming question. This model can then be used to rank a list of pre-authored answers. The model can be thought of as learning the coherence relationships between questions and answers. Our solutions for unsupervised dialogue models, specifically models described in section 4.6.5– 4.6.6, are a hybrid of such coherence models which are learned from in-domain example dialogues. We for the first time train and evaluate a *Cross-lingual Relevance Model* from dialogue transcripts with no additional manual effort of linking inputs with outputs as is otherwise required by the NPCEditor tool ([Leuski et al., 2006](#)).

2.3 Evaluating Dialogue Models

The final contribution of the thesis is towards rapid evaluation of the dialogue model. Most of the work on evaluating dialogue systems focuses on human-machine communication geared towards a specific task. A variety of evaluation metrics can be reported for such task-oriented dialogue systems. Dialogue systems can be judged based on the performance of their components like word error rate (WER) for ASR ([Jurafsky and Martin, 2000](#)), concept error rate or F-scores for NLU, understandability for speech synthesis etc. Usually the core component, the dialogue model - which is responsible for keeping track of the dialogue progression and coming up with an appropriate response, is evaluated indirectly. Different dialogue models can be compared with each other by keeping the rest of components fixed and then by comparing the dialogue systems as a whole. Dialogue systems can report subjective measures such as user satisfaction scores and perceived task completion. SASSI ([Hone and Graham, 2000](#)) prescribes a set of questions used for eliciting such subjective assessments. The objective evaluation metrics can include dialogue efficiency and quality measures.

PARADISE ([Walker et al., 2000](#)) was an attempt at reducing the human involvement in evaluation. It builds a predictive model for user satisfaction as a linear combination of some objective measures and perceived task completion. Even then the system needs to train on the data gathered from user surveys and objective features retrieved from logs of dialogue runs. It still needs to run the actual dialogue system and collect objective features and perceived task completion to predict user satisfaction.

Other efforts in saving human involvement in evaluation include using simulated users for testing ([Eckert et al., 1997](#)). This has become a popular tool for systems employing reinforcement learning ([Levin et al., 1997](#); [Williams and Young, 2007](#)). Some of the methods involved in user simulation

are as complex as building dialogue systems themselves (Schatzmann et al., 2007). User simulations are further required to be evaluated as how closely they model human behavior (Georgila et al., 2006) or as how good a predictor they are of dialogue system performance (Williams, 2007).

Some researchers have proposed metrics for evaluating a dialogue model in a task-oriented system. Henderson et al. (2005) used the number of slots in a frame filled and/or confirmed. Roque et al. (2006a) proposed hand-annotating information-states in a dialogue to evaluate the accuracy of information state updates. Such measures make assumptions about the underlying dialogue model being used like form-based or information-state based etc.

We are more interested in evaluating types of dialogue systems that do not follow these task-based assumptions: systems designed to imitate human-human conversations. Such dialogue systems can range from chatbots like Alice (Wallace, 2003b) and Eliza (Weizenbaum, 1966) to virtual humans used in simulation training (Traum et al., 2005). For such systems, the notion of task completion or efficiency is not well defined and task specific objective measures are hardly suitable. Most evaluations report the subjective evaluations for appropriateness of responses. Traum et al. (2004) propose a coding scheme for response appropriateness and scoring functions for those categories. Gandhe et al. (2006) propose a scale for subjective assessment for appropriateness.

Chapter 3

Rapid Development of Advanced Question Answering characters by Non-experts

The practical goal of this thesis is to enable rapid and cost-efficient prototyping of dialogue systems for Virtual Humans. Once the dialogue system architecture and the type of resources to be used have been determined, the cost of developing a dialogue system can be lowered by reducing the cost of building these specific resources. One approach toward reducing this cost is to allow *non-experts* to build such resources. By *non-experts*, we mean scenario authors who need not have any background in computational linguistics or any experience in building dialogue systems; although they can be an expert in the specific domain of interaction (e.g., tactical questioning, negotiation, etc.)

A typical system employing an information state based dialogue manager will need resources such as information state update rules and a collection of utterances annotated with dialogue acts to train the natural language understanding module. Authoring information state update rules requires expertise in computer science and computational theories of dialogue. Annotating utterances with dialogue acts needs expertise in linguistics (See Figure 1.3 and 3.5 for examples of utterances annotated with dialogue acts). Depending on the complexity of the dialogue act scheme, sometimes even experts find it difficult to annotate utterances with dialogue acts in a *consistent* manner.

Our approach toward making the authoring of dialogue systems accessible to *non-experts* is to use a *genre-specific* dialogue manager and an integrated authoring tool. If we use a genre-specific dialogue manager then we need to author the domain-independent information state update rules only once. This will allow building dialogue systems for several different domains without requiring expertise in rule authoring. This approach uses a *minimalist* dialogue act scheme that is complex enough to allow dialogue behaviors required for the specific genre, yet simple enough for *non-experts* to understand. Using such a dialogue act scheme along with an integrated authoring tool, we can have *non-experts* rapidly author dialogue system resources for several domains.

The work described here is performed for the genre of Advanced Question Answering Virtual Humans. In the following sections we elaborate more on the genre of advanced question answering characters and describe the dialogue system architecture used and the reasons for choosing it. This is followed by a detailed explanation of DomainEditor – an integrated authoring tool and the accompanying dialogue manager. Lastly we evaluate our authoring approach and show how 10 virtual human characters were built by authors with little or no experience in building dialogue systems. For one such character, Amani, we evaluate the dialogue act scheme being used by the authoring tool and also evaluate the resulting dialogue system.

3.1 Advanced Question Answering Characters

As introduced in Section 1.1, for simple question answering dialogue genre, the response to a question is an answer that addresses the question. This answer depends only on the question under discussion and is independent of the dialogue context (e.g., SGT Blackwell (Leuski et al., 2006) and SGT Star (Artstein et al., 2008), virtual soldiers who engage in simple question answering dialogues about U.S. Army). In contrast, advanced question answering characters may respond differently to the same question in different contexts.

Tactical Questioning dialogues are a special case of such advanced question answering dialogues. During Tactical Questioning, small-unit military personnel, usually on patrol, hold conversations with individuals to produce information of military value (Army, 2006b). Building Tactical Questioning characters that can play the role of a person being questioned has been an on-going project at the Institute for Creative Technologies. The simulation training environment is used to train military personnel in how to conduct such dialogues. Interviewers can employ several strategies to influence the characters to be cooperative and answer truthfully. Such strategies include building a rapport with the character, addressing their concerns, promising to do certain actions in their favor or pointing out the effects of non-cooperation.

Figure 3.1 shows one such example of a simulated tactical questioning dialogue. Here the trainee captain is interviewing a virtual human named Hassan. Throughout this chapter we will use examples from this domain. The scenario takes place in Iraq, where the U.S. authorities have built a marketplace and that is not being used by the locals. The goal for the trainee captain is to find out the reason by interviewing a local politician, Hassan. The trainee can use tactics such as complimenting Hassan in order to establish a rapport (e.g., utterances C5–H6). Hassan also has his own concerns about releasing sensitive information and needs certain promises from the captain. E.g., The question asked by the captain in utterance C9 is answered by Hassan in utterance H12.2 only after the captain has promised to offer a financial reward. We need advanced question answering dialogue systems to track such long-distance dependencies (e.g., C9–H12.2) as well as global social variables that can effect how the virtual character responds (e.g., level of rapport, number of offers promised, number of compliments made). For more examples of advanced question answering dialogues from other domains, see Figure 3.13 and 3.14. Figure 3.13 shows a dialogue where the virtual character Amani is being interviewed regarding a shooting incident she witnessed. Figure 3.14 is a sample dialogue with Avery, who wants to report a suspicious incident that took place at a U.S. Army base.

Advanced question answering virtual humans can also be used by psychology researchers as a low-cost alternative to human confederates during human subject testing. In such cases, the response given to a question from a human subject depends on variables directly under the researcher’s control (e.g., lying or not lying (Lane et al., 2010), replying with or without humor (Khooshabeh et al., 2011)).

3.2 Dialogue System Architecture

Our goal is to allow *non-experts* to rapidly author resources for advanced question answering dialogue systems. This goal, in turn, dictates the choice of dialogue system architecture. At the Institute for Creative Technologies, one such advanced question answering virtual human dialogue systems project, Tactical Questioning, has evolved through a few different architectures (Traum et al., 2008a). The first generation architecture was based on a simple question answering system (Leuski et al., 2006), where scenario designers were allowed to author questions, answers and links between these. The links, which could be many-to-many, indicate which answers are appropriate for which questions. Although the simplicity of the system allows *non-experts* in dialogue systems to author such scenarios, the architecture

C1	Captain	<i>greeting</i>	Hello
H2	Hassan	<i>greeting</i>	Hello
C3	Captain	<i>ynq</i>	Are you Hassan?
H4.1	Hassan	<i>yes</i>	Yes
H4.2	Hassan	<i>assert</i>	My name is Hassan.
C5	Captain	<i>compliment</i>	You have a nice home.
H6	Hassan	<i>response-compliment</i>	Thank you, but it was much more magnificent before. We can only repair so much.
C7	Captain	<i>whq</i>	Tell me about the market. I see there aren't very many people there and that doesn't seem typical for this area.
H8.1	Hassan	<i>grounding repeat-back</i>	So you want to talk about the market.
H8.2	Hassan	<i>assert</i>	The market is not being used because of the tax.
C9	Captain	<i>whq</i>	Ok, I'm trying to understand where the local taxation is coming from?
H10.1	Hassan	<i>grounding repeat-back</i>	So you want to talk about the taxes.
H10.2	Hassan	<i>elicit-offer</i>	I might tell you what you want if there was something in it for me.
C11	Captain	<i>offer</i>	We can offer you financial reward.
H12.1	Hassan	<i>response-offer</i>	That is very generous of you.
H12.2	Hassan	<i>assert</i>	Please understand, I collect taxes for my Imam. All in service to Allah.
C13	Captain	<i>whq</i>	And what is your Imam's name?
H14.1	Hassan	<i>grounding repeat-back</i>	So you want to talk about my Imam.
H14.2	Hassan	<i>elicit-offer</i>	My friend, if people find out that I tell you this, it would be a problem for me.
C15	Captain	<i>offer</i>	You will be anonymous. We will keep your anonymity.
H16.1	Hassan	<i>response-offer</i>	Ok, I trust you to keep this a secret.
H16.2	Hassan	<i>assert</i>	My Imam's name is Abdullah.

Figure 3.1: An example of advanced question answering dialogue with virtual human character, **Hassan**. *Captain* refers to the human trainee. Third column indicates dialogue act type.

suffers from the inability to maintain coherence over utterances that goes beyond just the adjacent ones (e.g, C9–H12.2 in figure 3.1).

The second generation architecture (Roque and Traum, 2007) made use of hand-authored rules for tracking affective variables, offers and threats made. These variables were used to compute a *compliance level*, which would dictate how the character would respond. There were three possible compliance levels – adversarial, reticent and compliant. The system's response was determined using text-to-text mappings similar to the first generation approach. But this architecture required developers to specify complete input-text to output-text mappings for all three compliance levels and worsened the authoring burden. Although this architecture could track changes in compliance throughout the dialogue it still could not handle the long-distance dependencies across sequences of utterances greater in length than two.

In order to engage in simple negotiation dialogues we need the ability to maintain such long-distance dependencies and global information state variables. [Gandhe et al. \(2008\)](#) describes the latest dialogue system architecture which is a *dialogue act based* architecture employing the information state model ([Traum and Larsson, 2003](#)). Figure 3.2 shows the dialogue system architecture along with the authoring tool. Such an architecture allows for the required advanced question answering behaviors. The dialogue manager has a set of modules for response generation, information state tracking, grounding and subdialogue tracking. The dialogue manager queries the domain knowledge base as required. In order to allow *non-experts* to author dialogue system resources we use an integrated authoring tool. The authoring tool can be used to collect resources for NLU, NLG, dialogue manager and populating the knowledge base. There have been efforts towards developing dialogue system authoring environments as summarized in section 2.1, but most of them require expertise in computer science and/or linguistics. The Hassan domain introduced earlier has been implemented in all three architectures ([Traum et al., 2007](#)).

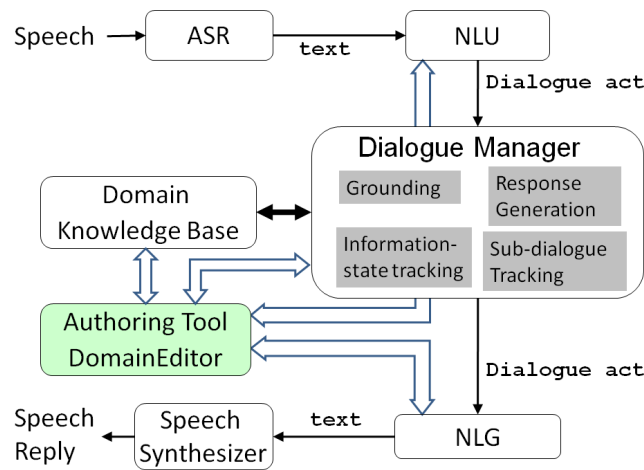


Figure 3.2: Architecture for the Tactical Questioning Dialogue System

3.3 Dialogue System Authoring

For dialogue act based architecture, the first two steps are corpus collection and dialogue act annotation. If the domain of interaction is relatively simple then the collection of dialogue corpus can be bypassed and the domain can be specified in a top-down fashion. Our authoring tool allows direct specification of the domain and also simplifies the corpus annotation so as to make it accessible to *non-experts*.

Dialogue act annotation can be viewed as a process of establishing correspondence between two different representations of an utterance: the surface text and the dialogue act. As with any such translation process there are two options – *generation* or *selection*¹. In the *generation* approach, we can compose the dialogue act, specifically the propositional content, from its basic elements. This approach allows the dialogue act to be a more detailed, faithful representation of the surface text. See Figure 1.3a for an

¹Here we focus on conversion to dialogue acts. For a similar discussion about conversion to surface text see section 4.3

example. But this approach has some drawbacks. First, this approach requires considerable expertise in linguistics. Second, the dialogue acts created by a composition procedure will have a lot of variation, too many unnecessary details and may not be *consistent* with the dialogue manager. Here *consistent* refers to a valid dialogue act that can be correctly handled by the dialogue manager. Third, the domain built using the propositional contents of the generated dialogue acts may be less coherent and the resources collected may be *incomplete*. E.g., We may have an utterance annotated as an answer but not have the corresponding question annotated because such a question was never encountered in the corpus. This results in dialogue manager having incomplete abilities such as the ability to answer questions it cannot understand or understand questions it cannot answer.

In the *selection* approach, we annotate surface text with a dialogue act that is selected from a set of valid dialogue acts. This approach requires annotators to coerce a surface text into one of the dialogue acts and may result in loss of information. There are advantages of this approach. First, with use of appropriate authoring tools, the *selection* approach affords time-efficient annotation. Second, since the approach allows selection from only a valid set of dialogue acts, all annotated dialogue acts are *consistent*. Third, a separate mechanism in the authoring tool can ensure that the set of all dialogue acts is *complete*. For our authoring process we use the *selection* approach for dialogue act annotation.

Our authoring process has two phases. In the first *top-down* phase, the authoring process begins by specifying the domain knowledge for the virtual character. We have designed a simple schema for specifying the domain knowledge which is easily understandable by *non-experts*. The scenario designers then collect dialogue system resources *consistently* and *completely* with the help of the authoring tool. Here *consistency* ensures annotations with only the valid dialogue acts and *completeness* refers to generating all dialogue acts that are relevant with respect to the character's domain knowledge and associating all of these dialogue acts to corresponding surface text.

The second phase is a *bottom-up* phase during which designers collect more dialogue data by having volunteers interact with the virtual human character which has been built during the first phase. The collected corpus can then be annotated with most appropriate dialogue acts. For some utterances this may require expanding the character's domain of knowledge. The authoring tool provides these necessary functions while ensuring *consistency* and *completeness*. During this phase the virtual human character is improved in an iterative manner by filling in the gaps in the domain knowledge of the character, making language understanding more robust and conducting more test runs with volunteers.

3.4 DomainEditor: Integrated Authoring tool

For the dialogue system architecture described in section 3.2, the resources required are: the domain knowledge of the virtual character, a set of dialogue acts that the dialogue manager needs to reason with, examples of surface text annotated with these dialogue acts and the information state update rules. In order to reduce the development cost, most of these resources should be authored by *non-experts* – authors with no expertise in linguistics or computer science and no previous experience in building dialogue systems.

We have developed an Integrated Authoring tool – **DomainEditor**, that allows *non-experts* to author the domain knowledge for the virtual character and surface text examples for dialogue acts. All relevant dialogue acts are automatically constructed by the authoring tool using a *genre-specific minimalist* dialogue act scheme. The information state update rules used by the dialogue manager still require considerable expertise to author. Our approach is to build a library of *genre-specific* but *domain-independent*

pre-defined dialogue behaviors (viz. question-answering, offer subdialogues, greeting, closing, grounding, etc.) and let the scenario designers select which ones they want for their character. The *domain-dependent* update rules can be authored by the *non-expert* scenario designers using the policy authoring functionality of the tool.

For advanced question answering dialogues, we need the ability to model simple negotiations over when to release certain sensitive information. The dialogue manager maintains a model of emotions and compliance which are updated as the dialogue proceeds. In compliant mode, the character may elicit certain offers from the interviewer before answering questions regarding the sensitive information. Whereas in adversarial mode, the character may choose to lie in response to these questions. DomainEditor allows scenario authors to mark certain information elements as sensitive and modify some of the policies regarding when to release this information.

There are cases where we would like to build several characters that can be questioned about the same incident. E.g., Multiple witnesses of a shooting incident at the marketplace will have a considerable overlap in their domain knowledge. One of the requirements for the authoring tool is the ability to re-use the existing authored content across different characters. Our tool allows for such re-use of the domain knowledge along with all the dialogue acts and the language associated with it.

Figure 3.3 shows a screenshot of our authoring tool. It has three horizontal panels. The topmost panel is used for editing the domain knowledge level. The middle one allows authors to view all dialogue acts and select one of them. The bottom panel allows editing of the surface text corresponding to the chosen dialogue act.

3.4.1 Domain Knowledge Level

Domain knowledge is created as a four level hierarchy. The highest level is the *characters*, the conversational participants in the domain, who can be speakers and addressees of utterances and dialogue acts. In the Hassan domain there are two characters viz. the trainee interviewer (called *player*) and Hassan. Each character knows about a set of *objects*. These objects can be of different types such as person (Hassan, Imam), location (market) or abstract concept (tax). Objects can be associated with different types of knowledge bits. *Attributes* can be used to describe various properties of the object. These attributes can take on *values* which can be marked as true or false (to be used as lies). Objects of type person can also have representations for *actions* they can perform (e.g., offers, threats). Actions are not further specified with values. Another aspect crucial for tactical questioning is social behavior for building rapport. This behavior is represented by special actions such as *compliments* and *insults* whose arguments are the objects being complemented. Person-type objects can also have *goals* but currently these are used just as a talking point. In future we plan to connect goals with attributes and actions.

We use a simple XML representation of these domain knowledge aspects, for ease of use across modules of the system at both domain specification and run-time. Figure 3.4 shows parts of the domain specification used for the Hassan scenario. The Hassan domain has two characters – *player* and *hassan*. The character *hassan* knows about a few objects (e.g., *hassan* (himself), *tax*). The *tax* object is further described by an attribute *collector*, which has two values – one true value, *hassan* and one false value, *tax-collecting-soldier*, to be used as a lie). The character *player* knows about an object *player* (himself) and the actions that the *player* object can perform such as offers (*give-money*, *provide-secrecy*) and complimenting objects (*house*).

A basic proposition is a triple ⟨object, attribute, value⟩. Assertions of such propositions constitute the answers and queries for the value field of such propositions form the basis for questions. The simple object, attribute, value scheme supports a simple question answering ability. These additional aspects of domain (e.g., actions, compliments) can be used to provide advanced question answering abilities by creating domain-dependent dialogue policies as described in section 3.5.2.

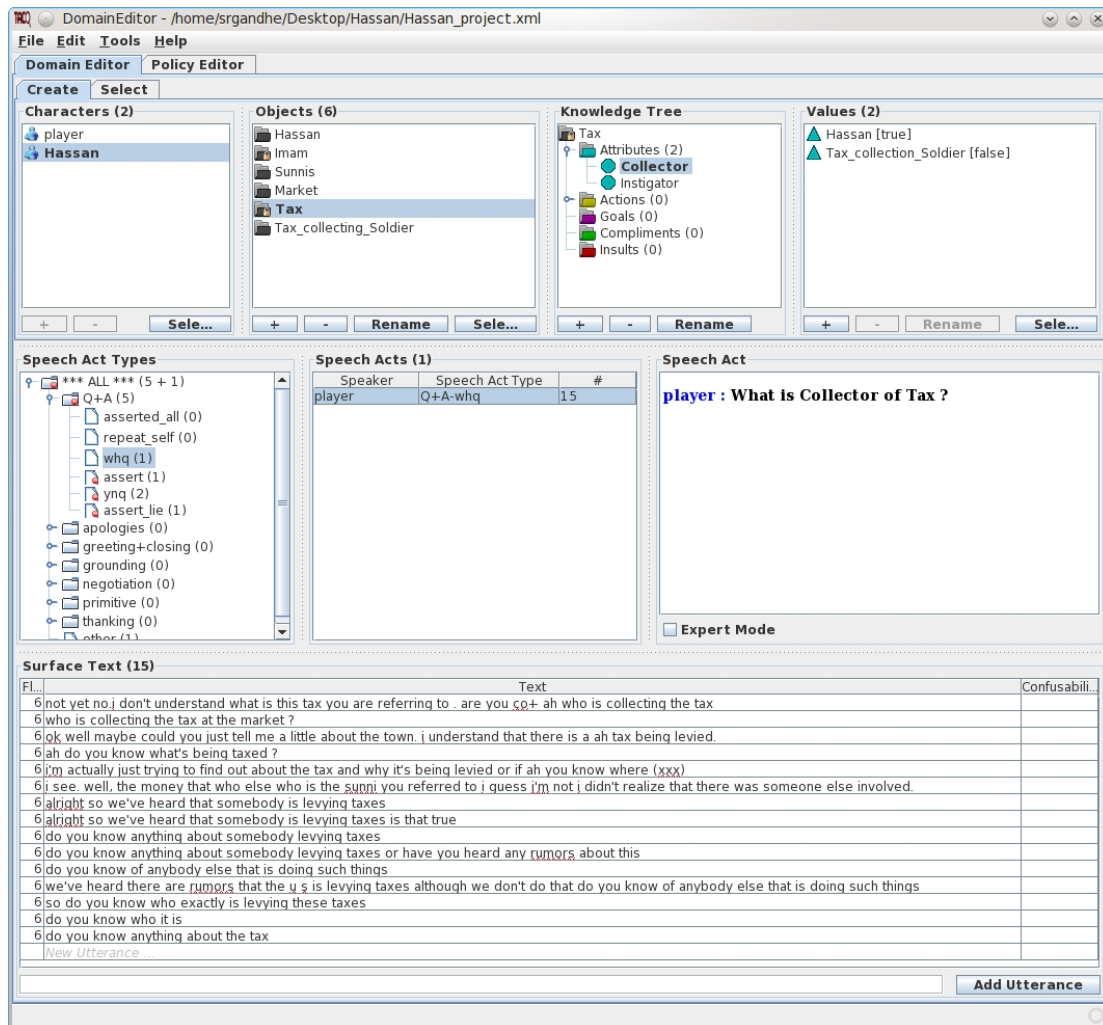


Figure 3.3: DomainEditor: An Integrated Authoring tool for designing the domain, and specifying the utterances that map to various dialogue acts.

The topmost panel of the authoring tool (see Figure 3.3) shows the domain creation aspects, where, moving from left to right, authors can add or delete characters, objects, object contents (attributes, actions, compliments, etc.) and values. The tool automatically constructs XML like that shown in Figure 3.4. This top most section is also used to filter the set of dialogue acts shown in the middle panel of the GUI.

3.4.2 Dialogue Act Level

Once the domain is defined, it needs to be linked up with the language that will be used to refer to it. Dialogue acts form the middle level in this link, having domain aspects as their contents and being identified directly as the interpretations of language utterances.

```

<domain name="hassan">
  <character name="hassan">
    <object name="hassan" type="person">
      <attribute name="role">
        <value truthiness="true">middle-man</value>
      </attribute>
      <actions>
        <offer name="cooperate"/>
      </actions>
    </object>
    <object name="tax" type="abstract">
      <attribute name="collector">
        <value truthiness="true">hassan</value>
        <value truthiness="false">tax-collecting-soldier</value>
      </attribute>
    </object>
    ...
  </character>
  <character name="player">
    <object name="player">
      <actions>
        <offer name="give-money"/>
        <offer name="provide-secrecy"/>
      </actions>
      <compliments>
        <object name="house"/>
      </compliments>
      ...
    </object>
  </character>
</domain>

```

Figure 3.4: Aspects of the *Hassan* domain

Our dialogue manager reasons about several standard types of dialogue acts, including *assertions*, *yn-questions*, *wh-questions*, *offers*, *threats*, *compliments* and *insults*. Following [Core and Allen \(1997\)](#) we have dialogue acts with forward-function – *elicitations* and with backward-function – *responses* for most of the acts. Figure 3.5 shows our XML representation of some of the dialogue acts, which contain a speaker (one of the characters), an act-type, and the contents.

All dialogue acts are automatically created from the domain representation as per Algorithm 3.1. E.g., all $\langle \text{object}, \text{attribute}, \text{value} \rangle$ triples known by a character can serve as the contents of an *assert* with that character as the speaker. Likewise, any $\langle \text{object}, \text{attribute} \rangle$ pair known by another character can be queried with a *wh-question* addressed to that character. We also generate some generic dialogue acts that are customary in human-human conversations like *greeting* and *closing*, that are not tied to any specific domain content. Grounding acts like *repeat-back* (see utterance H8.1 & H10.1 in figure 3.1) and *request-repair* are also generated ([Roque, 2009](#)). *Offtopic* is a special dialogue act specifically designed to handle out-of-domain dialogue acts from the player. Figure 3.15 shows a complete list of dialogue act types and dialogue behaviors supported by them.

The middle panel of the authoring tool shown in Figure 3.3 allows selection from among the full set of dialogue acts. The left pane allows selection of the type of dialogue act; the middle pane lets one select individual dialogue acts; the right pane shows the content of the selected dialogue act. Instead of showing the detailed XML representation, we use pseudo-natural language – generated using templates. E.g., We use a template like “*Attribute of Object is Value*” for the *assert* dialogue act type. Such a template would generate “Collector of Tax is Hassan” as a representation for the *hassan.assert* dialogue act shown in Figure 3.5. Authors are also allowed to replace the automatically generated template text with a selection from the surface text examples associated with the selected dialogue act. The detailed XML representation of the dialogue act is also available under the *expert mode*.

hassan.assert

```
<dialogue_act speaker="hassan">
  <primitive_speech_act>
    <assertion>
      <object name="tax">
        <attribute name="collector">
          <value>hassan</value>
        </attribute>
      </object>
    </assertion>
  </primitive_speech_act>
</dialogue_act>
```

Indeed, you might say that I collect the taxes.

player.offer

```
<dialogue_act speaker="player">
  <primitive_speech_act>
    <offer name="give-money"/>
  </primitive_speech_act>
</dialogue_act>
```

We can offer you financial reward.

hassan.elicit-offer

```
<dialogue_act speaker="hassan">
  <elicit>
    <primitive_speech_act>
      <offer name="give-money"/>
    </primitive_speech_act>
  </elicit>
</dialogue_act>
```

I might tell you what you want if there was something in it for me.

Figure 3.5: Sample dialogue acts automatically generated from the *Hassan* domain along with example utterances.

3.4.3 Surface Text Level

The natural language understanding module converts the surface text to a dialogue act. The NLU uses a statistical language modeling text classification technique (Leuski and Traum, 2008). In case the closest dialogue act match falls below a threshold, an *unknown* dialogue act is passed on to the dialogue manager. The Natural language generation module which converts a dialogue act to surface text works in a similar fashion but in reverse direction. Both NLU and NLG require a training corpus of sample utterances linked to dialogue acts, which can be produced using the authoring tool.

This annotated corpus is generated during both the *top-down* and the *bottom-up* phase. The authoring tool shown in Figure 3.3 supports this via links between natural language texts in the bottom pane, and dialogue acts in the middle pane. During the *top-down* phase, for each dialogue act from the character, the author can add one or more options for the character to realize this act. Likewise, for the player dialogue acts, the author can list many possible ways for the player (interviewer) to produce this act. During the *bottom-up* phase, utterances collected from test runs of volunteers interacting with virtual human are annotated with dialogue acts.

The task of generating a training corpus for NLU and NLG can be time consuming. It is mitigated by allowing utterances to be linked only to a dialogue act drawn from a specific set of automatically

Algorithm 3.1 Algorithm for generation of dialogue acts from domain specification

```
for all speaker ∈ characters do
  /* Primitive dialogue acts */
  for all obj ∈ objects under speaker do
    for all atr ∈ attributes under obj do
      for all val ∈ values under atr do
        ADD assertions (speaker, obj, atr, val)
      end for
    end for
    for act ∈ actions under obj do {ADD actions - offers/threats (speaker, obj, act)}
    for goal ∈ goals under obj do {ADD goals (speaker, obj, goal)}
    for compl ∈ compliments under obj do {ADD compliments (speaker, obj, compl)}
    for insult ∈ insults under obj do {ADD insults (speaker, obj, insult)}
    ADD groundingDAs (speaker, obj)
  end for
  /* Dialogue acts that relate to other characters */
  for all char' ∈ (characters \ speaker) do
    for all obj' ∈ objects under char' do
      /* Forward-looking dialogue acts */
      for all atr' ∈ attributes under obj' do
        ADD whq (speaker, obj', atr')
        for all val' ∈ values under atr' do
          ADD ymq (speaker, obj', atr', val')
        end for
      end for
      for act' ∈ actions under obj' do {ADD elicit-action (speaker, obj', act')}
    end for
    /* Backward-looking dialogue acts */
    for act' ∈ actions under obj' do {ADD response-offer, response-threat (speaker, obj', act')}
    for compl' ∈ compliments under obj' do {ADD response-compliment (speaker, obj', compl')}
    for insult' ∈ insults under obj' do {ADD response-insult (speaker, obj', insult')}
    ADD groundingDAs (speaker, obj')
  end for
end for
/* Generic dialogue acts */
ADD greetings, closings, accept, reject, refuse-answer, ack, offtopic ...
end for
```

generated dialogue acts which is *consistent* and *complete*. It is easier to choose a dialogue act for an utterance rather than construct one from scratch. Utterances have multiple functions and can be marked up with multiple dialogue acts. But for simplicity, we annotate only the most salient dialogue act. This may result in parts of utterances not having any representation in the selected dialogue act. As an example consider the dialogue act shown in Figure 3.6 along with some corresponding sample utterances. In utterance 2, the clause “so that you could do other things that will better benefit allah.” does not have any representation in the dialogue act. By avoiding the construction of the dialogue act from scratch and focusing on the most salient part, we can facilitate and speed up the annotation process of such utterances. This produces *consistent* annotations which by design will be handled correctly by the dialogue manager. Some of the utterances in Figure 3.6 are a result of corpus collection through user testing of the virtual

human dialogue system during the bottom-up phase. If available, roleplays or WoZ sessions can also be annotated in a similar fashion. If the non-represented parts of these utterances are deemed important, then the domain specification can be expanded to include those using the tool shown in Figure 3.3. The tool will also automatically generate dialogue acts which will be appropriate elicitations/responses to the new additions, thus ensuring *completeness*.

```
<speech_act speaker="player">
  <primitive_speech_act>
    <offer name="protect-hassan"/>
  </primitive_speech_act>
</speech_act>
```

- 1 ***I promise you that you will not receive any harm** for giving me this information.*
- 2 *Well I can also help you in other ways and **we can protect you** so that you could do other things that will better benefit allah.*
- 3 *Well, if you could help us, the perhaps **we could put you in protection. and offer you protective custody** because if your people are being taxed unfairly, then you're being taxed unfairly as well too and perhaps we can help.*
- 4 *Sure I understand, as I said, **I can make you safe** ah if you're able to share information with me. but ah hopefully that will be enough.*

Figure 3.6: A sample dialogue act along with the corresponding surface text utterances. The most salient part of these utterances which matches with the dialogue act is highlighted.

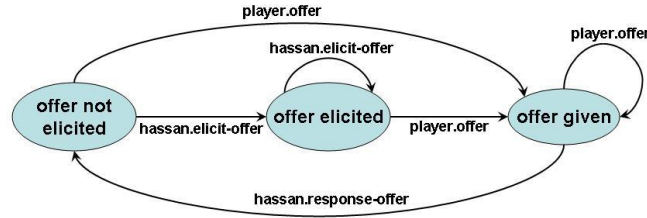
3.5 Dialogue Manager

We use an information-state based dialogue manager (Larsson and Traum, 2000) and this information-state is in part based on the conversational game theory (Lewin, 2000). The main responsibilities of the dialogue manager are to update the information state of the dialogue and use it to select the contents of the response. The dialogue manager gets input dialogue acts from the NLU and outputs dialogue acts to the NLG. It decomposes the dialogue acts in order to update the information state.

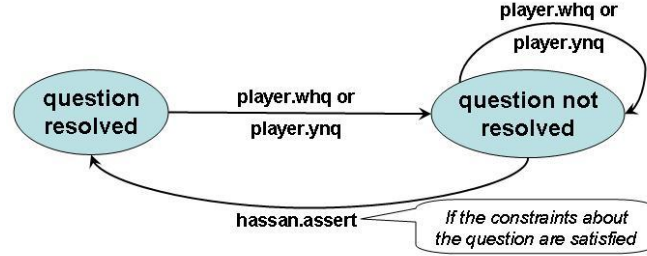
The information state update rules describe grammars for conversational game structure and are written as state charts. We are using State Chart XML (SCXML), a W3C working draft (Barnett et al., 2008), for describing state charts. SCXML allows for explicit data models that can be manipulated by executable code. This code can be triggered on entry or exit from a state or during a transition. As pointed out by Kronlid and Lager (2007), all these features make it viable to implement an information-state based dialogue model using SCXML².

We have defined a set of networks for each type of game/subdialogue. The states in these networks model the virtual character's conversational obligations (Traum and Allen, 1994) and the state transitions represent the dialogue acts. Figure 3.7 shows networks that handle the offer subdialogue and the question-answering subdialogue. Each state indicates the state of the character's obligations. E.g., For the question-answering subdialogue (Figure 3.7b), the initial state 'question-resolved' indicates that there are no pending obligations. The outgoing arcs from the currently active states with *player* as the speaker denote all possible dialogue acts that can be handled as input from the user. E.g., In the 'question-resolved' state the dialogue manager can consume a *player.whq* or *player.ynq* dialogue act and activate the 'question-not-resolved' state, indicating a pending obligation to address the question asked. Outgoing

²We used the apache commons SCXML implementation. [<http://commons.apache.org/scxml>]



(a) offer subdialogues



(b) question-answer subdialogues

Figure 3.7: State charts for *Hassan* domain.

arcs with the virtual character as a speaker indicate ways to address that obligation. E.g., The dialogue act *hassan.assert* addresses the obligation by answering the question. Some of the transitions in these networks are conditional and depend on the data model configuration (i.e. information-state). These conditions can be domain-dependent and can be authored by scenario designers using DomainEditor.

We have authored subdialogue networks for greeting, compliment, insult, question-answering, offer, threat, pre-closing, closing and grounding subdialogues. Consistent with our design approach of allowing *non-experts* to rapidly build the dialogue systems, the scenario developer is expected to select from such a set of subdialogues/games for a given domain. We have already identified a finite set of games that are particularly relevant for the advanced question-answering dialogue genre. Still an expert user can manually author subdialogue networks from first principles if needed.

3.5.1 Functions of Dialogue Manager

Updating Information State The dialogue manager performs several functions; the most important of which is to keep track of the information state. The information state represents the dialogue history (context) by maintaining information sufficient to generate an appropriate response for that context and understanding the input related to the context (Traum and Larsson, 2003). This information state gets updated after consuming an input dialogue act and after producing an output dialogue act. In our architecture, the information state keeps track of active states from all the subdialogue networks. Each subdialogue maintains appropriate information to conduct that particular subdialogue (e.g., The question-answer network remembers the question under discussion. The offer network stores the offer under discussion.). The dialogue manager keeps track of social commitments made by the virtual character or by the interviewer through assertion dialogue acts.

C9	Captain	<i>whq</i>	Ok, I'm trying to understand where the local taxes	taxes
			taxation is coming from?	
H12.2	Hassan	<i>assert</i>	Please understand, I collect taxes for my Imam. All in service to Allah.	taxes
C13	Captain	<i>whq</i>	And what is your Imam's name?	Imam
H16.2	Hassan	<i>assert</i>	My Imam's name is Abdullah.	Imam
C17	Captain	<i>unknown</i>	<i>non-understanding</i> (ASR, NLU errors)	Imam
H18	Hassan	<i>request-repair-object</i>	Are you talking about my Imam?	Imam
C19	Captain	<i>yes</i>	yes	Imam
H20	Hassan	<i>request-repair-attribute</i>	What about my Imam do you want to know?	Imam
C21	Captain	<i>unknown</i>	<i>non-understanding</i> (ASR, NLU errors)	Imam
H22	Hassan	<i>giveup-repair</i>	I may not know what you are asking about.	Imam

(a)

H16.2	Hassan	<i>assert</i>	My Imam's name is Abdullah.	Imam
C17	Captain	<i>unknown</i>	<i>non-understanding</i> (ASR, NLU errors)	Imam
H18	Hassan	<i>request-repair-object</i>	Are you talking about my Imam?	Imam
C19	Captain	<i>no</i>	no	-
H20	Hassan	<i>request-repair</i>	Can you repeat your last question?	-

(b)

Figure 3.8: Example dialogue excerpts showing topic tracking & grounding functions of the dialogue manager. The last column indicates the *topic* of conversation after processing the corresponding utterance.

It also tracks which offers/threats have been given, which offers/threats are to be elicited, number of offers/threats, number of compliments/insults, dialogue length, etc.

Formulating response The dialogue manager builds an output dialogue act based on the information state and the domain knowledge of the character. The labels for all outgoing arcs from currently active states with character as the speaker are candidates for the output dialogue act. If these outgoing arcs are specified with any conditions these conditions are evaluated based on current configuration of the information state. If there are more than one possible output dialogue acts then these acts are ordered based on a fixed priority. The default policy is to generate all possible output dialogue acts as part of a composite response turn. The turn is released when there are no more output dialogue acts possible.

Topic tracking & Grounding The dialogue manager keeps track of the current topic of conversation which is the most recent *object* associated with previous dialogue acts. The topic is fed back to NLU to bias the interpretations of future utterances towards dialogue acts about this object/topic. There is a special set of networks for tracking topic and implementing the grounding behavior similar to [Roque and Traum \(2009\)](#). If the input dialogue act is *unknown*, which indicates non-understanding, the dialogue manager first confirms the current topic of conversation and then requests the interviewer to repeat or rephrase their utterance. If non-understanding persists even after two consecutive attempts of repairing the dialogue then the repair process is abandoned and the interviewer is informed about it. Figure 3.8 shows sample dialogues illustrating this behavior.

Advanced Dialogue Behavior Keeping track of the information state enables the dialogue manager to handle underspecified *wh-questions* as “Tell me more about the soldier.” The dialogue manager responds by asserting a previously un-released piece of information about the relevant *object* “soldier” (See Figure 3.14 utterances P11..A12). It can also detect utterances that are asking the same question and point it out to the interviewer (See Figure 3.14 utterances P15..A16.2).

Emotions, Social variables, Personality The dialogue manager also keeps track of the emotional state, social variables, personality variables for the virtual character. For Hassan, these variables are feels-respected, respects-interviewer, social-bonding and fear (Roque and Traum, 2007). Based on these variables the character’s compliance level is determined as adversarial, reticent or compliant. This compliance level influences what kind of reply will be given. E.g., when adversarial, the character may choose to lie in response to questions, if a lie is available. For Amani, Rushforth et al. (2009) modeled variables such as assertiveness, modesty, honesty, trust, positive emotion, activity, etc. which affect the personality of the character.

3.5.2 Policy Authoring

Although the network structures themselves are currently hand-authored by experts as SCXML documents, some of the constraints for these networks can be authored by *non-experts* using the authoring tool.

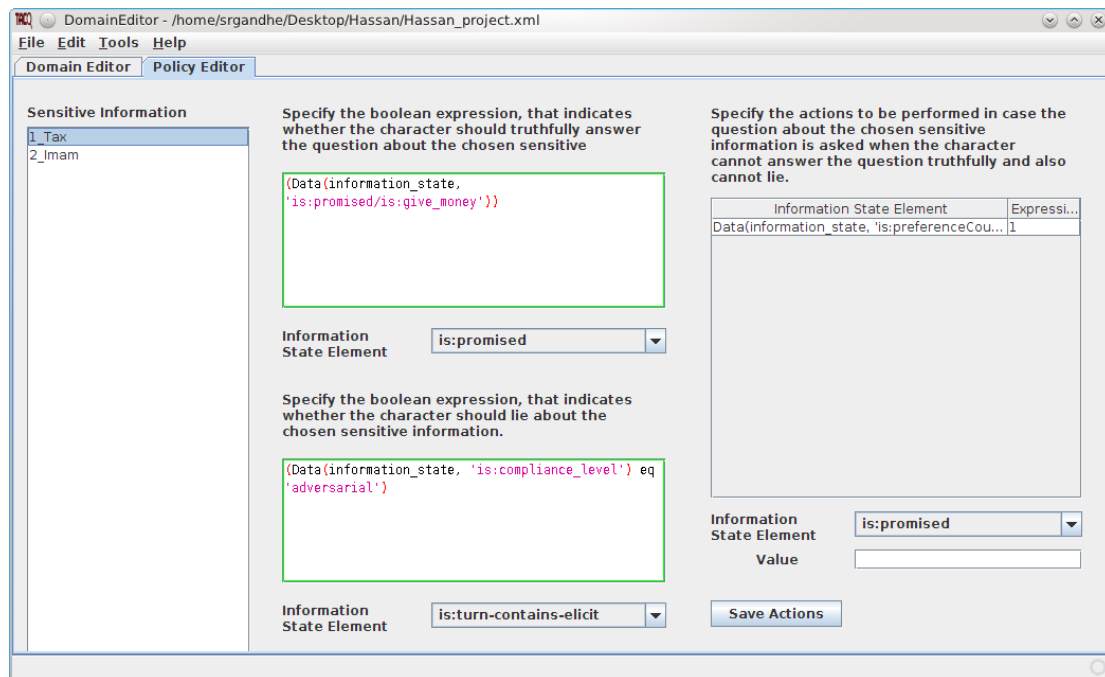


Figure 3.9: The authoring tool can be used to specify the conditions for question-answering network.

Figure 3.9 shows the policy editing pane. The leftmost pane lists domain elements that are marked as sensitive information. Information can be marked as sensitive at the level of an object or a specific attribute of an object. For every piece of sensitive information the author can provide conditions under

which the character should truthfully answer the question or lie about it. These conditions can be any boolean expressions formed by using the information state elements which can be chosen from a drop-down list. A question about this sensitive information will not be answered until the corresponding condition evaluates to true. In case the character cannot answer the question truthfully then the condition for lying is evaluated. If the condition for lying evaluates to true the character lies about the sensitive information, otherwise the actions specified in the rightmost pane are executed. As an example, consider the dialogue policy shown in Figure 3.9. Any *yn-question* or *wh-question* about the *object* “Tax” will not be answered unless the player has promised the *offer* of “give-money”. If the player has not yet promised to give money then Hassan will lie if he is in *adversarial* mode. Otherwise the dialogue manager will set a preference for the *offer* “give-money”, which in turn will result in the next move from Hassan being an *elicit-offer*.

P1	Player	<i>whq</i>	question resolved, offer not elicited Ok I’m trying to understand where the local taxation is coming from?
H2.1	Hassan	<i>grounding</i>	question not resolved, offer not elicited So you want to talk about the taxes.
H2.2	Hassan	<i>elicit-offer</i>	I might tell you what you want if there was something in it for me.
P3	Player	<i>offer</i>	question not resolved, offer elicited We can offer you financial reward.
H4.1	Hassan	<i>response-offer</i>	question not resolved, offer given That is very generous of you.
H4.2	Hassan	<i>assert</i>	question not resolved, offer not elicited Please understand, I collect taxes for my Imam. All in service to Allah.
			question resolved, offer not elicited

Figure 3.10: Example dialogue showing the currently active states for the networks in Figure 3.7.

Figure 3.10 shows a sample dialogue corresponding to the above policy. Hassan is in *compliant* mode and the player has not yet offered to “give money”. When the player asks a sensitive question (utterance P1), the dialogue manager decides whether to answer this question truthfully by evaluating the condition specified in the authored policy (see Figure 3.9). Since the “give-money” offer is not yet promised and the character is not in *adversarial* mode, Hassan neither gives a truthful answer nor lies. Instead, Hassan sets the preference for “give-money” offer and chooses to start the offer subdialogue by eliciting that offer (utterance H2.2). After utterance P3 the constraints are met. Hassan can then respond to the offer (*hassan.response-offer* – utterance H4.1) thus completing the offer subdialogue and answer the question (*hassan.assert* – utterance H4.2) thus resolving the question under discussion and completing the question-answer subdialogue.

3.6 Evaluation

Here we present the evaluation of our authoring approach. The authoring tool, DomainEditor, generates all dialogues acts automatically based on a fixed *genre-specific minimalist* dialogue act scheme and the character’s domain knowledge. DomainEditor supports editing of the character’s domain knowledge but not the dialogue act scheme. Although it is possible to change the dialogue act scheme for different domains, such an adaptation requires expertise in software development and dialogue act theories. Since

the tool only allows annotating an utterance with a dialogue act that has been automatically generated, the chosen dialogue act scheme could be a limiting factor. So first, we present the evaluation of the dialogue act scheme we used for advanced question-answering genre. Next we evaluate our authoring tool and provide evidence that *non-experts* can build advanced question-answering characters using the authoring tool (Gandhe et al., 2011). We provide some details about the domains and the development times for the virtual humans that were developed by *non-experts*. Finally we present an overview of how these virtual characters have been successfully used in various applications.

3.6.1 Evaluation of the Dialogue Act Scheme

To test whether a dialogue act scheme allows for advanced dialogue behavior, we need to estimate the *coverage* of the dialogue act scheme. *Coverage* refers to the portion of user utterances that can be represented by one of the dialogue acts from the set of automatically generated acts. The user utterances are collected during test runs of the dialogue system. To verify whether the annotation task is simple enough, we need to measure inter-annotator agreement for dialogue act annotation. We have conducted a dialogue act annotation study for a virtual character, Amani (Artstein et al., 2009b, 2011). In this domain, Amani has witnessed a sniper shooting at U.S. troops. The trainee, who plays the role of a U.S. Army platoon leader can interview Amani to find out the details about what she saw and the identity and location of the shooter. The trainee may need to make some assurances to Amani in order to get her to cooperate in this investigation. (See Figure 3.13 for an example interaction).

The dialogue act annotation study was conducted in three phases. During the first phase we conducted pilot testing of the Amani scenario and improved the dialogue act scheme. Next we conducted field testing of Amani at the U.S. Military Academy, Westpoint. The test corpus collected was roughly split in two parts. During the second phase, one part of this corpus was annotated with dialogue acts. The set of dialogue acts automatically generated by DomainEditor is a function of the dialogue act scheme used as well as the virtual character’s authored domain. During this second phase we found that the coverage can be improved by expanding the domain of conversation (*bottom-up phase*). A scenario designer expanded the Amani domain using DomainEditor. Finally in the third phase the remaining part of the Westpoint corpus was annotated. during all three phases of the annotation study a subset of utterances was annotated by multiple annotators to serve as a corpus for judging inter-annotator agreement. Table 3.1 summarizes the results.

Corpus	No. of player utterances	No. of player DAs	Coverage	Reliability α		
				sample size	individual DA	in/out of domain
Pilot studies at ICT	224	113	50%	224	0.49	0.38
Westpoint (original)	768	143	62-68%	90	0.49	0.33
Westpoint (expanded domain)	799	287	72-76%	110	0.63	0.39

Table 3.1: A summary of dialogue act annotation reliability and domain coverage for different corpora at different developmental stages for Amani.

During the pilot phase, a total of 224 unique player’s utterances were matched by 3 annotators to the closest dialogue act. Utterances which did not match an appropriate existing dialogue act were marked with the special *unknown* dialogue act. There were a total of 113 possible player dialogue acts to choose from, 53 of which were selected at least once by one of the annotators. We found the coverage to be around 50% and the inter-annotator agreement as measured by Krippendorff’s α to be

0.49 when calculated on individual dialogue acts³. This value is substantially above chance, but fairly low compared to accepted standards. In this setting the dialogue acts include the illocutionary force as well as the associated propositional content. We also calculated the inter-annotator agreement on the task of deciding whether an utterance is marked with the special *unknown* dialogue act or not, i.e. whether an utterance can be coerced into one of the existing dialogue acts or whether a new dialogue act needs to be created by extending the domain knowledge of the character. The low value for this, $\alpha = 0.38$, indicates that it is a difficult decision for annotators.

More detailed analysis of the annotations suggested that some of the disagreements were due to unclear guidelines and will not result in a significant impact on the system performance. E.g., Whether a question of the form *Do you know...* or *Can you tell...* should be treated as a *yn-question* or *wh-question*? In both cases the dialogue system will respond by answering the question. The analysis also revealed some gaps in the coverage of our dialogue acts scheme, such as the absence of under-specified questions which ask about an object without specifying an attribute, as in *Tell me more about the sniper*. Since such questions are very common, constituting nearly 12% of our pilot corpus, we added corresponding dialogue acts to the dialogue act scheme.

We conducted field testing of Amani at U.S. Military Academy, Westpoint. A total of 34 participants interviewed Amani. These were the users from our target population. Each participant interviewed Amani twice, resulting in a corpus of 68 dialogues consisting of 1854 player utterances. This corpus was divided in two parts. The first part, referred to as Westpoint (original) with a total of 768 unique player utterances, was then annotated with dialogue acts in a similar fashion but with improved guidelines. The domain was also modified based on the observations from the pilot study and there were a total of 143 player dialogue acts to choose from. Although the inter-annotator reliability remained the same the coverage improved.

After the Westpoint (original) corpus was annotated, one of the annotators went through all the utterances marked with *unknown* dialogue act and expanded the domain in order to accommodate as many of them as possible. We performed the same dialogue act annotation study on the rest of the held-out corpus, referred to as Westpoint (expanded domain), and consisting of 799 unique player utterances. The coverage and inter-annotator reliability both improved substantially. Although the decision of whether an utterance is out-of-domain is still difficult, the overall reliability improves as the domain coverage improves. The analysis shows that with improved guidelines and extensions, our dialogue act scheme can produce reliable annotations and adequately represent around 76% of actual player utterances. For more detailed analysis see (Artstein et al., 2009b, 2011).

With the current dialogue act scheme, we have found it is possible to conduct meaningful dialogues with the virtual humans in the advanced question-answering genre. Despite the best efforts of perfecting the dialogue act scheme and expanding the domain for a specific virtual human, during run-time some of the player's utterances will still be mapped to the *unknown* dialogue act. This may happen due to ASR or NLU errors. The dialogue manager is equipped to handle these using the grounding network. The dialogue manager attempts to confirm the topic of the conversation and then asks the user to repeat or rephrase. Other possible strategies to handle *unknown* dialogue act include taking initiative by providing related information about the current topic of conversation if in *compliant* mode or give an *offtopic* response.

³Krippendorff's α (Krippendorff, 2004) is a generalized measure of inter-rater agreement, similar to the more familiar K. For a detailed discussion of inter-rater agreement coefficients, see (Artstein and Poesio, 2008).

3.6.2 Evaluation of the Integrated Authoring Tool, DomainEditor

The authoring tool, DomainEditor has been used for creating several tactical questioning characters (viz. Hassan, Amani, Ali Sadat, Sean Avery, Lisa) at ICT. Besides tactical questioning, DomainEditor has also been used to develop advanced question-answering characters that can be used as confederates for human subject testing (viz. Victor, Amber, Bradley). For these characters, the designers found the top-down structured authoring more approachable than the un-structured authoring process from the previous generation (Leuski et al., 2006). Additionally, the resulting dialogue systems have support for dialogue behaviors such as topic tracking & grounding. Recently, DomainEditor has also been used to build characters that can engage in negotiations (viz. Jabbar, Sadik). Figure 3.11 shows a list of characters along with the information about their authors, corresponding scenarios and the amount of dialogue system resources that were collected.

Hassan was implemented in previous architectures and was ported to the new architecture as the authoring tool was being developed. The rest of the ten characters were authored by *non-experts*. **Amani** was initially developed by a *non-expert* as a tactical questioning character. We conducted a pilot user testing for this character as well as user testing at U.S. Military Academy (USMA) Westpoint where we had access to the target users for tactical questioning systems. A total of 34 cadets interviewed Amani twice and practiced their interviewing skills. This corpus allowed us to identify and fix the deficiencies in our initial dialogue act schema (Artstein et al., 2009b, 2011). **Sean** and **Avery** represent two episodes involving the same character PFC Sean Avery. PFC Avery has witnessed an incident involving a U.S. Army soldier and an accomplice smuggling something suspicious on the army base. These characters were developed by the same author over 3 months. The **Avery** character is more complex and includes dialogue policies for deciding whether or not to lie about certain information based on what has happened in the dialogue. **Lisa**, another character in the extended PFC Sean Avery scenario, was developed by another *non-expert* author during the following summer. Lisa is the accomplice involved in the smuggling incident. **Ali Sadat** was developed by a USMA cadet, who used his subject matter expertise to effectively circumscribe the character's domain knowledge. One of the guidelines for tactical questioning is to fill out a SALUTE (Size, Activity, Location, Uniform, Time, Equipment) report (Army, 2006a). By aligning the structure of the domain with the structure of the SALUTE report the author was able to develop the domain of interaction in a short time.

Besides tactical questioning, our tool has been used by psychology researchers to build advanced question-answering characters which can be used in their experimental methodologies. These virtual characters provide a consistent experience compared to human confederates and can be controlled precisely by the system designer. **Victor** and **Amber** are two such characters that were developed to teach how to use verbal cues for deception detection. They can answer questions truthfully or deceptively depending on the mode in which they are being operated. DomainEditor is well suited for creating such characters. A total of 35 participants interacted with these two characters (Lane et al., 2010). For this study, the input interface was typed text with an optional multiple-choice between suggested similar questions while the virtual humans responded with speech performed by animated bodies. **Bradley** was another such character designed to study social influence of humor (Khooshabeh et al., 2011). A total of 54 participants had conversations with either a humorous or non-humorous version of Bradley using typed text interface for both input and output.

Recently the authoring tool – DomainEditor and the dialogue manager described here was used by *non-experts* to develop negotiation characters. **Jabbar** is one such virtual human who starts the negotiation by making unrealistic demands for money. The interviewer who is playing the role of a U.S. Army lieutenant has the goal of maintaining good relations with locals and can use tactics like active listening and establishing rapport by getting to know more about Jabbar and his family. Jabbar will reveal his true interests only if the interviewer has successfully established a rapport. **Sadik** is




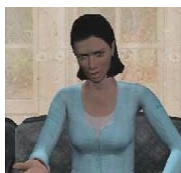


Hassan 	Developers Scenario Dev Time Size	A Team of dialogue system researchers U. S. Army has built a marketplace in Iraq, which is not being used by the locals. Hassan can be questioned to find out why as he knows something about the illegal tax collection going on at the market. 1 year Player 108 DAs 187 utterances Hassan 102 DAs 129 utterances
Amani 	Developers Scenario Dev Time Size	Sarah Ali, Undergrad student in Computer Science (non-expert in dialogue systems) Amani has witnessed a recent shooting in the marketplace. The interviewer is to question her to find out the identity, location and description of the shooter. See Figure 3.13 for a sample interaction. 4 months Player 113 DAs 681 utterances Amani 89 DAs 98 utterances
Victor 	Developers Scenario Dev Time Size	Stephen Michael, Grad student in Psychology (non-expert in dialogue systems) Victor was one the two characters developed for a tutoring system (Lane et al., 2010) which is designed to teach verbal cues for deception detection. Victor is witness to a bombing at a local abortion clinic. He can operate in two modes truthful or deceptive. 4 months (along with Amber) Player 240 DAs 2317 utterances Victor 170 DAs 170 utterances
Amber 	Developers Scenario Dev Time Size	Stephen Michael Amber, who has witnessed a shooting, is the second character from the deception detection tutoring system. 4 months (along with Victor) Player 240 DAs 1792 utterances Amber 169 DAs 152 utterances
Sean 	Developers Scenario Dev Time Size	Aly Taylor, Undergrad student in Communication (non-expert in dialogue systems) PFC Sean Avery has witnessed a fellow soldier smuggling something suspicious on a U.S. Army base. He can be questioned about what he saw, who the soldier was and who was the accomplice. 3.5 months (along with Avery) Player 151 DAs 707 utterances Sean 103 DAs 172 utterances
Avery 	Developers Scenario Dev Time Size	Aly Taylor This is the same character PFC Sean Avery interviewed again after the accomplice has been apprehended. Meanwhile PFC Sean Avery has realized that the soldier involved in the smuggling was from his platoon and now wants to cover up the incident. He may choose to lie and will need more coercion in form of threats & offers. See Figure 3.14 for a sample interaction. 3.5 months (along with Sean) Player 193 DAs 811 utterances Avery 147 DAs 256 utterances

Figure 3.11: Various Virtual Human characters that have been created using DomainEditor. 45





Ali Sadat 	Developers	CDT Jonathan Hoey, U.S. Military Academy Westpoint cadet, undergraduate student in Systems Engineering (non-expert in dialogue systems)		
	Scenario	Ali Sadat is a shop keeper in Afghanistan and knows about Taliban activities regarding IEDs.		
	Dev Time	2 weeks		
	Size	Player	182 DAs	658 utterances
Bradley 	Developers	Peter Khooshabeh, PhD in Psychology (non-expert in dialogue systems)		
	Scenario	Bradley is a fellow crew member on a space ship which has crash landed on the moon. He is the inventory specialist and can be interviewed in order to prioritize a list of 15 items in this Lunar Survival task. The character is part of an study to investigate how humor affects social influence.		
	Dev Time	4 months		
	Size	Player	288 DAs	1207 utterances
Lisa 	Developers	Garrett Langhauser, undergraduate student in Industrial Systems Engineering (non-expert in dialogue systems)		
	Scenario	This scenario is an extension from the previous Sean Avery scenario. Lisa Jefferson is PFC Ryan Benton's girlfriend and was the civilian accomplice who was involved in smuggling the suspicious item on U.S. Army base. She can be questioned to find out more about the smuggling incident and her motive.		
	Dev Time	4 weeks		
	Size	Player	168 DAs	1062 utterances
Jabbar 	Developers	Garrett Langhauser		
	Scenario	Jabbar is the head of an Afghan Apple co-operative who claims he has been cheated by the U.S. Army and demands 2 million USD as compensation. The interviewer needs to reason with Jabbar and build a trusting relationship in order to uncover exactly what happened.		
	Dev Time	7 weeks		
	Size	Player	244 DAs	1555 utterances
Sadik	Developers	James Clark, U.S. Military Academy Westpoint cadet, undergraduate student in Systems Engineering (non-expert in dialogue systems)		
	Scenario	Sadik is an Afghan National Police Chief, who is in charge of an area where Taliban have recently set fire to an Afghan supply truck. The interviewer playing the role of a captain in U.S. Army has to convince Sadik to take control of the situation. During the negotiation the interviewer may need to use tactics like active listening & inquiry, coming up with creative options, etc.		
	Dev Time	3 weeks		
	Size	Player	179 DAs	826 utterances
		Sadik	134 DAs	141 utterances

Figure 3.11: Various Virtual Human characters that have been created using DomainEditor.

another negotiation character which can be used for training persuasion skills. The interviewer needs to convince Sadik to take charge of a mission. The interviewer can use tactics like active listening, asking why instead of what and coming up with alternate options. The negotiation behavior was authored by making certain domain information sensitive and writing policies about when to truthfully answer the sensitive questions and when to lie. We've found these mechanisms to be adequate for developing simple negotiation characters.

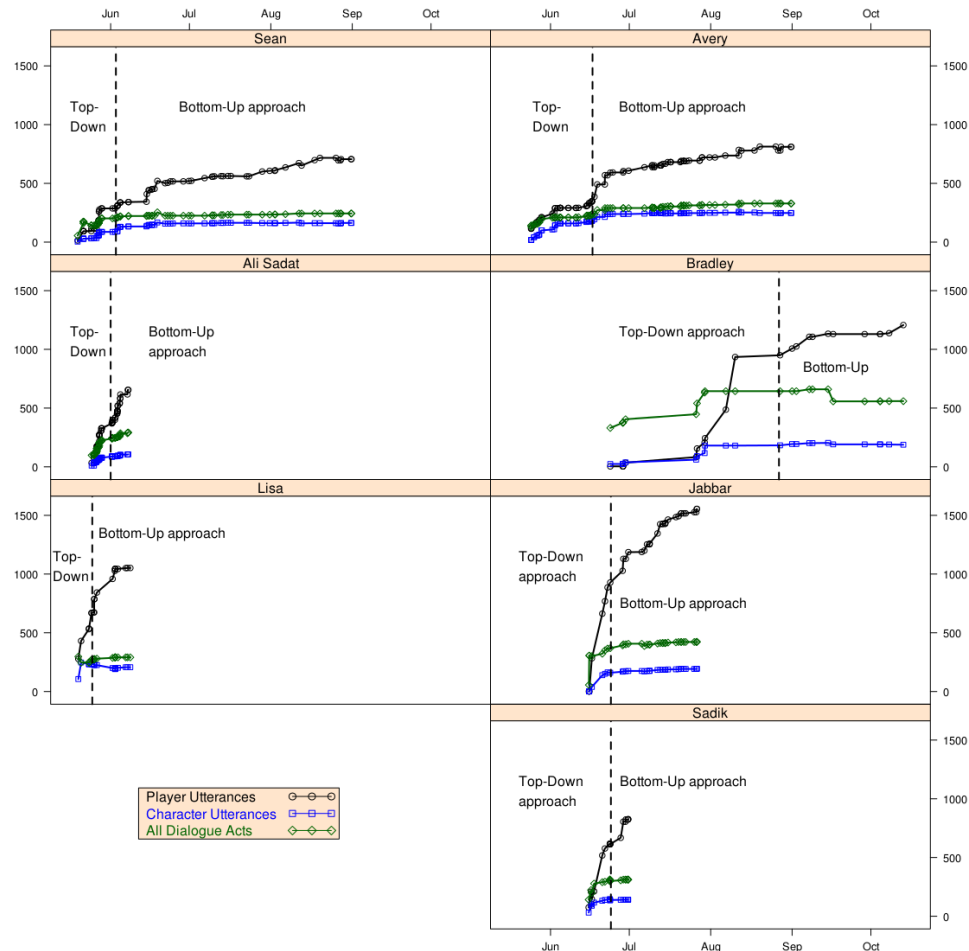


Figure 3.12: Amount of dialogue system resources collected across time for several characters which were authored using DomainEditor by non-experts.

Figure 3.12 shows the authoring progress of seven such characters which were developed by *non-experts* during summer 2010 & 2011. It shows that non-experts can use the authoring tool to build virtual human dialogue systems in a small amount of time. In fact, Ali Sadat was developed in only 14 days by a Westpoint cadet.

The authoring process for these characters has two phases. The first phase begins with a top-down process which includes defining the character's domain knowledge first and then authoring the surface

Question: The responses are on a scale of 1 (negative) to 7 (positive)	Old architec- ture (Roque & Traum, 2007) (n=8)	New architec- ture (Gandhe et al., 2008) (n=7)
How satisfied were you with your interview with Hassan?	3.38	4.14
How would you rate your performance in questioning Hassan?	2.75	5.00 *
How would you rate Hassan as an interviewee?	3.88	3.86
How well do you think Hassan understood your speech?	3.88	4.29

Table 3.2: Subjective evaluation for the character, Hassan with old and new architectures and corresponding authoring processes. The values reported here are the means of responses to different questions. * indicates statistical significance ($p < 0.01$) based on wilcoxon rank-sum test.

text for all relevant dialogue acts. The growth in number of dialogue acts represents the growth in character’s domain knowledge. As can be seen from figure 3.12, the domain reaches a stable level relatively early in the process. Most of the domain authoring occurs during this phase. Scenario designers author one or two utterances for each of the character’s dialogue acts for some variability. Substantially more examples are authored for player dialogue acts in order to ensure robust NLU performance. The second phase is a bottom-up phase which involves collecting a dialogue corpus by having volunteers interview the virtual human character that has been built. The utterances from this corpus can then be annotated with the most appropriate dialogue act. It can be seen that this second phase is responsible for a rapid growth in player utterances. It can also lead to minor domain expansion and small increase in character utterances.

3.6.3 Resulting Virtual Human Dialogue Systems

We compared this new authoring process and the resulting virtual human characters with characters from the previous generation architecture. Compared to the old authoring, our authoring process takes a more structured approach – starting with defining the domain knowledge for the character and then providing surface text examples for automatically generated dialogue acts. We need to evaluate whether such added structural constraints lead to any adverse effects on the resulting virtual human dialogue system. Hassan is a character that has been implemented with the new third generation architecture (Gandhe et al., 2008) as well as the previous second generation architecture (Roque and Traum, 2007). Table 3.2 presents subjective evaluation of the character, Hassan for these two architectures. It shows that the performance of the resulting dialogue systems is comparable. Both versions of Hassan were developed by a team of dialogue system researchers in approximately one year. The development time for Hassan based on the third generation architecture includes the time required for developing the DomainEditor and the corresponding dialogue manager.

We also evaluated the virtual human characters that were authored by non-experts using the authoring tool. Table 3.3 shows the results of subjective evaluation of these virtual human characters as compared to the expert-authored Hassan. The results suggest that *non-expert* authored characters can perform at par with the expert authored characters.

These virtual humans developed by *non-experts* were not just developed as a proof-of-concept but were used in the field. They have been successfully used in research studies leading to publications. Amani was field tested at U.S. Military Academy, Westpoint where 34 participants, enrolled in a negotiation class, practiced their skills with the virtual human. Virtual Characters Amber and Victor have

Question: on a scale of 1 (negative) to 7 (positive)	Hassan (n=10)	Amani (n=33)	Avery (n=17)	Lisa (n=9)	Bradley (n=95)
How appropriately did the character respond to what you were saying?	4.00	3.24	3.71	4.22	4.98
How believable was the character in its role?	3.55	3.09	3.53	4.22	4.23

Table 3.3: Evaluation of the resulting Virtual Human dialogue systems for characters built using our authoring process and the integrated authoring tool, DomainEditor.

been used in human subject testing where a total of 35 participants interviewed the characters (Lane et al., 2010). Bradley has participated in research study by having conversations with 95 participants to date (Khooshabeh et al., 2011). Besides the virtual characters themselves, the scenario and corpora collected have also lead to research in the field of dialogue systems. Specifically the Amani scenario and collected corpus has been used for evaluation of speech recognition systems (Yao et al., 2010), automated dialogue evaluation (Margaretha and DeVault, 2011) and dialogue policy learning (DeVault et al., 2011b).

3.7 Conclusion

In this chapter, we have explained how advanced question answering virtual humans can be rapidly developed by *non-experts*. The advanced dialogue behaviors expected from these characters dictates the use of an information-state based dialogue manager. This choice in turn implies that the resources to be collected include the domain knowledge of the character, all the relevant dialogue acts for this domain and examples of surface text for these dialogue acts. The integrated authoring tool, DomainEditor provides a simple ontology for defining the domain and hides the complexity involved in authoring dialogue acts. The authoring tool can be used by *non-experts* to collect the required resources *consistently* and *completely*. This tool supports both *top-down* and *bottom-up* phases of character development. To date, 10 virtual humans have been developed by *non-experts* using this authoring tool and the development time can be as low as two weeks.

The practical goal of this thesis is to enable rapid and cost-efficient prototyping of dialogue systems for Virtual Humans. Our approach toward reducing this cost is to allow *non-experts* to build the required resources. In this chapter we have shown that, by using a *genre-specific* dialogue manager and an integrated authoring tool, *non-experts* can rapidly build advanced question answering characters.

		both not greeted, compliment replied, question resolved, offer not elicited, unknown responded, track topic
P1	<i>greeting</i>	hello amani greeted
A2	<i>greeting</i>	Hello both greeted
P3	<i>compliment</i>	you have a wonderful house. compliment given
A4	<i>response-compliment</i>	This house used to be much nicer before.
P5	<i>whq</i>	compliment replied can i ask you a few questions about the shooting incident yesterday? question not resolved
A6	<i>assert</i>	Go on if you can be quick about it. question resolved
P7	<i>unknown</i>	i would like to assure you that you will be protected. do request repair
A8	<i>request-repair</i>	You were not clear in what you said. Say it again. unknown responded
P9	<i>offer</i>	i promise to keep you safe. offer given
A10	<i>response-offer</i>	You promise me safety. offer not elicited
P11	<i>whq</i>	did you know who did the shooting? question not resolved, repeat back
A12.1	<i>repeat-back</i>	So you want to talk about the man who did the shooting. question not resolved, track topic
A12.2	<i>assert</i>	I have heard that he is called Saif. question resolved
P13	<i>whq</i>	do you know where he lives?
A14	<i>elicit-offer</i>	You are asking for a lot of information. If you expect me to tell you that you must offer me secrecy.
P15	<i>offer</i>	i will keep this a secret.
A16.1	<i>response-offer</i>	You have assured me of secrecy.
A16.2	<i>assert</i>	I believe he hides on the second floor of Assad's shop. I know he is in there. but when i go in the shop i don't see him. And i have no idea where he is now.
P17	<i>whq</i>	do you know anything about the shooter's daily activities?
A18	<i>elicit-offer</i>	As you know, I am a teacher. We are not provided the luxuries of great wealth. One hundred of your dollars would go a long way to help my family make it through these hard times. question not resolved, offer elicited
P19	<i>reject</i>	sorry i'm not authorized to give you any money. question not resolved, offer not given
A20	<i>refuse-answer</i>	There is not much i wish to say about that. question resolved, offer not elicited

Figure 3.13: A sample dialogue with Amani. The first line shows currently active states (shown in **bold**) for greeting, compliment, question-answer, offer, unknown and grounding subdialogue networks. In subsequent lines only the change in the active state is shown. P is the player (human trainee) and A is Amani. In Amani domain *Shooter's name*, *location* and *daily routine* are marked as sensitive information. The *name* will be released only if *safety* is promised. Since *safety* is already offered in utterance P9, it is not elicited after utterance P11. Information regarding *daily routine* will be released only if *give-money* is promised. In utterance P19 player rejects the elicitation of this offer which results in Amani refusing to answer (utterance A20).

P1	Player	<i>ynq</i>	Are you Sean Avery?
A2.1	Avery	<i>yes</i>	Yes
A2.2	Avery	<i>assert</i>	My name is PFC Sean Avery.
P3	Player	<i>whq</i>	Tell me what you saw.
A4	Avery	<i>assert</i>	The soldier got out of the car and took a package from the back seat.
P5	Player	<i>whq</i>	Do you know who the soldier was?
A6	Avery	<i>assert-lie</i>	I have no clue what the soldier's name is.
P7	Player	<i>threat</i>	If you don't tell me your privileges will be revoked.
A8	Avery	<i>response-threat</i>	I don't want to be on restriction.
P9	Player	<i>whq</i>	Who was the soldier?
A10	Avery	<i>assert</i>	The soldier's name is Ryan Benton.
P11	Player	<i>whq</i>	Tell me more about the soldier.
A12	Avery	<i>assert</i>	The PFC is in my platoon and one of my closest friends.
P13	Player	<i>whq</i>	What was inside the package?
A14	Avery	<i>assert</i>	He was carrying a puppy.
P15	Player	<i>whq</i>	What was inside the package?
A16.1	Avery	<i>self-repeat</i>	As I said earlier,
A16.2	Avery	<i>assert</i>	The suspicious package contained a puppy.
P17	Player	<i>whq</i>	Whose idea was it to smuggle the puppy on the base.
A18	Avery	<i>elicit-offer</i>	Will you give the soldier a less severe punishment if I talk?
P19	Player	<i>accept</i>	yes
A20.1	Avery	<i>response-offer</i>	That's comforting, I wouldn't want him to get thrown off base or anything.
A20.2	Avery	<i>assert</i>	I think the soldier told his girlfriend it wouldn't be a big deal to sneak a puppy on base.

Figure 3.14: A sample dialogue with **PFC Sean Avery** character. *Player* refers to the human trainee. Third column indicates dialogue act type.

Dialogue Behavior	Dialogue Act type	Description/Example
Question-Answer	whq	A wh-question (<i>E.g., Who was the soldier that you saw?</i>)
	ynq	A yes-no question (<i>E.g., Did you see Ryan Benton this morning?</i>)
	assert	An truthful assertion generally in response to a question. (<i>E.g., The soldier's name was Ryan Benton.</i>)
	assert-lie	A false assertion, when the agent wants to lie. (<i>E.g., I do not know who the soldier was.</i>)
	refuse-answer	A refusal to answer that addresses the obligation put forth by a pending question without answering the question. (<i>E.g., I cannot answer that.</i>)
	yes	Yes
	no	No
	asserted-all	A special act which conveys that no further new information can be communicated about the object under discussion. (<i>E.g., I told everything I know about the soldier.</i>)
	repeat-self	A special act which conveys that the agent has already disclosed a particular assertion about <Object, Attribute, Value>. (<i>E.g., I think I already told you this.</i>)
Offer/Threat	elicit-offer elicit-threat	An utterance that generally results in an offer/threat being made in response. (<i>E.g., Will you give the soldier a less severe punishment if I talk?</i>) (<i>E.g., What are you going to do arrest me?</i>)
	offer threat	An Offer (<i>E.g., I will be lenient.</i>) (<i>E.g., I can issue a court-martial.</i>)
	response-offer response-threat	A response to a specific offer. (<i>E.g., Thats comforting, I wouldn't want him to get thrown off base or anything.</i>) (<i>E.g., I don't want to be court-martialed.</i>)
	accept	Accept the offer. (<i>E.g., I accept.</i>)
	reject	Reject the offer. (<i>E.g., I cannot accept these terms.</i>)
Greeting	greeting	Initiating a conversation. (<i>E.g., Hello</i>)
Pre-closing	pre-closing	Initiating a closing for the conversation. (<i>E.g., I need to leave now.</i>)
Closing	closing	Closing (<i>E.g., Goodbye.</i>)

Figure 3.15: Types of dialogue acts and the dialogue behaviors supported by them.

Dialogue Behavior	Dialogue Act type	Description/Example
Goal	goal	A statement informing the goals of the agent (E.g., <i>I am here to report an incident that occurred today morning.</i>)
Ack	ack	Acknowledgment (E.g., <i>Okay.</i>)
Compliment	compliment	Compliment
	response-compliment	Response to a specific compliment.
Insult	insult	Insult
	response-insult	Response to a specific insult.
Apology	apology	Apology (E.g., <i>Sorry</i>)
	response-apology	Response to apology (E.g., <i>Never mind.</i>)
Thanks	thanks	Thanks (E.g., <i>Thanks.</i>)
	response-thanks	Customary response to thanks (E.g., <i>You're welcome.</i>)
Topic-shift	set-topic	Used to explicitly set the topic of conversation. (E.g., <i>Let's talk about the soldier you saw in the morning.</i>)
	repeat-back	Used to explicitly confirm the current topic of conversation. (E.g., <i>So you want to talk about the soldier.</i>)
Repair & Grounding	request-repair-object	Request clarification about which topic/object to discuss. (E.g., <i>What do you want to talk about?</i>)
	request-repair-attribute	Request clarification about which attribute of the object to discuss. (E.g., <i>What about the soldier do you want to know?</i>)
	request-repair	Request for repair (E.g., <i>I did not understand that. Can you say that again?</i>)
	giveup-repair	After trying to repair conversation for several times, giveup the repair process. (E.g., <i>I am sorry I do not understand what you said.</i>)
	repeat	(E.g., <i>Pardon me.</i>)
	unknown	A special dialogue act when the NLU fails to interpret the incoming utterance.
	offtopic	A special dialogue act used as a reply to handle an out-of-domain utterance. (E.g., <i>I am having some trouble hearing, just returned from shooting practice.</i>)

Figure 3.15: Types of dialogue acts and the dialogue behaviors supported by them.

Chapter 4

Unsupervised Dialogue Models

In chapter 3, we presented an approach towards developing a virtual human dialogue system based on *dialogue act* architecture. It employs a genre-dependent dialogue manager and an integrated authoring tool that allows *non-experts* to collect required resources at a lower cost. But when the genre of interaction is different we still need experts to author additional information state update rules and extend dialogue act schema. Also it is not always possible to author the domain specification directly in a top-down manner. In such cases, collecting an in-domain dialogue corpus is the best way to start. Such corpus collection activity is very useful as it can help understand the actual scope of the language being used by the dialogue participants. In some cases an in-domain dialogue corpus (Roleplays and WoZ) may already be available.

In this chapter, we present flexible dialogue models that can be bootstrapped from in-domain human-human dialogue corpus. These models primarily work at the *surface text* level and do not require any rule writing or corpus annotations. Resources such as annotations for information state, dialogue acts, topics, etc. can be incrementally added to the dialogue model as they become available. In section 4.1 we introduce the testbed domain SASO-ST (Traum et al., 2005) which was used to evaluate these models. Section 4.2 presents a few assumptions used in building a dialogue system employing these dialogue models. Section 4.3 presents two approaches for response formulation: *generation* and *selection*. Our models use the selection approach. In section 4.4 we describe a method to evaluate the viability of the *selection* approach for a given dialogue corpus and in section 4.5 we present our selection criterion and the rationale behind it. Section 4.6, presents the dialogue models we have implemented including two baselines models. Section 4.7 presents an evaluation of these models in *dynamic context* and *static context* settings. We discuss some issues regarding the dialogue models we have implemented and the experimental results in section 4.8. Finally we list other possible applications of our dialogue models in section 4.9.

4.1 SASO-ST Testbed

At USC’s Institute for Creative Technologies, researchers have developed several prototype virtual human characters used for simulation training. SASO-ST (Traum et al., 2005) is one such environment, involving a prototype of a training environment for learning about negotiating with people from different cultures and with different beliefs and goals. In the first scenario, the trainee acts as an Army Captain negotiating with a simulated doctor (see Figure 4.1). The trainee’s goal is to convince the doctor to move his clinic to another location. The captain can offer help in moving the clinic and some other perks like medical supplies and equipment.



Figure 4.1: Virtual Human for Doctor character from SASO-ST scenario.

In order to investigate this domain, and build resources for the system, we collected a corpus of role-play dialogues and Wizard of Oz (WoZ) dialogues. Role-play dialogues feature more free-form human face to face interaction. Figure 4.2 shows a fragment of a typical role-play dialogue. On the other hand, the WoZ interactions are constrained by allowing the wizard playing the role of doctor to choose from a limited set of replies. Figure 4.3 shows a fragment of a WoZ dialogue. We have collected 23 roleplay dialogues and 13 WoZ dialogues. The dialogues last for 40 turns on average. There are a total of 703 captain turns and 707 doctor turns in this corpus resulting in a total of 29471 words. The contrast between the roleplays and WoZ data can be seen in the average length of the doctor’s turns - for roleplays it is ~ 19 words, whereas for WoZ it is ~ 8 words. The captain’s turns are not much affected (~ 28 words in roleplays and ~ 25 in WoZ).

4.2 Dialogue System Architecture

Here we use a *surface text based* dialogue system architecture as described in Figure 1.4. We are interested in building flexible dialogue models that operate at surface text level and do not require any annotations or rule authoring. Such a model should result in a functional dialogue system based on the only resource of un-annotated in-domain human-human dialogue transcripts available from roleplays or WoZ. As additional resources (e.g., information state annotations) become available, the models should have the ability to incrementally incorporate the resources.

In order to build such dialogue systems, we make a few simplifying assumptions:

Dialogue Genre The models presented here are for two party conversational dialogue systems for virtual humans. We have built dialogue models for a negotiation scenario, specifically SASO-ST.

Modality The input modality is restricted to verbal signals – typed text or recognized speech. In case of speech as input, we ignore the prosody information and only focus on speech transcription obtained from ASR. We also ignore other non-verbal inputs such as gesture, body position, head nods etc. For output modality, the model generates the response in the form of plain surface text. This surface text is then converted to synthesized speech (Hunt and Black, 1996; Aylett et al., 2006) along with the appropriate non-verbal behaviors (Lee and Marsella, 2006) using standard components from the VHToolkit¹.

¹<http://vh toolkit.ict.usc.edu/>

Doctor	yes what is it i've got a lot of patients in the back . what can i do for you .
Captain	how are you doing sir , uh my name's captain (xx) , how are you today ?
Doctor	uh well , i could be better , i've got a lot of patients in the back , uh we just had uh FIVE of them come in from the LAST bombing ? so , what can i do for you .
Captain	okay i know you're very busy so i'll get straight to what i came here to talk to you about . right now , with our estimate , this is a very unsecure area . and what we'd like to do sir is uh secure and stabilize your patients as soon as possible and move you out of this area so we can move you to a more secure location .
Doctor	my PATIENTS are stable right NOW . and , i i don't understand why you're coming in here , to tell me to move patients out of here , from a clinic that's been here for almost a YEAR . and now i have to move my patients ?

Figure 4.2: A sample role-play dialogue in SASO-ST domain.

Doctor	you are the threat i need protection from you.
Captain	you do not need protection from us you need protection from the insurgents that are killing innocent people in this area sir
Doctor	it is already unsafe
Captain	i know it is unsafe but we 're working to make it safer an example of our works is helping you move this clinic to a safer neighbourhood we just need your cooperation to
Doctor	you are the problem your bombs are killing these people
Captain	no sir it is not our bombs thats killing these people its the insurgents that are killing these people if you 'd simply let us help yo us this would benefit you and us by moving your clinic to a safer environment so you can more adequately help these people here
Doctor	ok are you going to attack
Captain	i cannot say exactly when and if we are going to attack i can only say that this area is getting more dangerous by the day and we need to help you relocate your clinic as soon as possible in fact within the next
Doctor	it sounds serious captain i will think about it

Figure 4.3: A sample Wizard-of-Oz dialogue in SASO-ST domain.

Turn taking Turns strictly alternate between the virtual human character and the human trainee (user). Alternatively, the strict turn taking assumption can be relaxed by using a separate turn-taking model which decides which speaker has the next turn.

Functionality Typically a dialogue manager is responsible for several functions such as progressively tracking the information state of the dialogue, providing context for interpretation of input utterances, formulating a response utterance, etc. The dialogue models we build have a single primary function – *formulating an appropriate response given a dialogue context*. These models are not required to track the information state. Hence these models will not be able to provide functions that require tracking information state such as explaining the virtual character’s behavior during an after action review, perceiving changes in its environment which is generally accomplished by changing certain information state elements, etc.

In the next three sections, we will present the decisions made regarding formulating a response and the reasons for it.

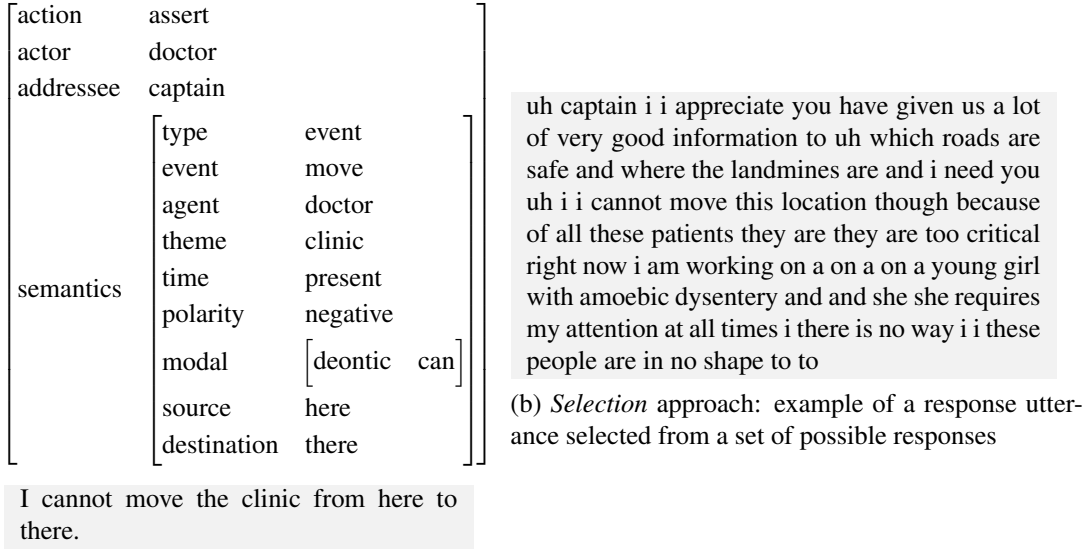
4.3 Formulating a Response: Generation Vs Selection

There are two main approaches toward formulating a response utterance in dialogue. One is the *selection* approach, in which the task is to pick the appropriate response utterance from a corpus of possible utterances. The other is the *generation* approach, in which the response utterance is dynamically assembled using some composition procedure, such as grammar rules to convert information from semantic representations and/or contextual information from previous utterances (information state). Our unsupervised dialogue models employ the *selection* approach.

The *selection* approach requires a set of possible response utterances which can be obtained from an in-domain human-human dialogue corpus. Such a resource can be collected by non-experts. The *generation* approach generally requires more analytical effort to devise a good set of grammar rules that cover the range of desired sentences but do not admit undesirable or unnatural sentences. Also the generation approach requires the input to be a dialogue act which includes a detailed semantic representation of the contents of the response. Thus generation approach requires considerably more expertise in computational linguistics as compared to the selection approach.

The generation approach has the advantage of a more compact representation for a given generative capacity. But for any finite set of sentences produced, the selection approach could perfectly simulate the generation approach. Moreover, since the selection approach chooses a response utterance from the utterances that have been observed in human-human dialogues, the response can be a complex, human-like sentence without requiring much detailed analysis. When the output is not just text but presented as speech, the system may easily use recorded audio clips rather than synthesized speech. This argument also extends to multi-modal performances, e.g., using artist animation, motion capture or recorded video for animating virtual human characters rather than procedural animation. Often one is willing to sacrifice some generality in order to achieve more human-like behavior than is currently possible from generation approaches. Figure 4.4 shows an example response utterance from each of these approaches with roughly the same meaning. The response generated by generation approach exactly matches the semantics expressed in the response dialogue act, no more no less. It can only be as expressive as the corresponding dialogue act. In contrast, the response chosen by selection approach has all the features a human-like response would have (e.g., disfluencies, repetitions, personality).

The working assumption used in the *selection* approach is that a **virtual human character can conduct a dialogue where the response is formulated by selecting an appropriate utterance from the corpus rather than constructing one from abstract representations**. There are many instances where



(a) *Generation* approach: example of a response utterance generated using a grammar from a response dialogue act

(b) *Selection* approach: example of a response utterance selected from a set of possible responses

Figure 4.4: An example illustrating the contrast between *Generation* and *Selection* approach.

the selection approach has been successfully used for dialogue modeling. In fact, the WoZ data collection process often employs the *selection* approach, where a wizard selects an appropriate utterance from a set of carefully crafted utterances. The selection approach has been used for a number of dialogue agents, including question-answering characters at ICT (Leuski et al., 2006; Artstein et al., 2009a; Kenny et al., 2007), FAQ bots (Zukerman and Marom, 2006; Sellberg and Jönsson, 2008) and web-site information characters (e.g., <http://www.alaskaair.com/Jenn>, <http://www.virtuoz.com>).

4.4 Viability of the Selection Approach

The selection approach presents two challenges for finding an appropriate utterance:

- *Is there a good enough utterance to select?*
- *How good is the selection algorithm at finding this utterance?*

We now formalize the notion referenced to by the first question. We perform a study of existing dialogue corpora to establish the theoretical maximum performance of the *selection* approach to simulating human dialogue behavior in unseen dialogues. Our methodology is to examine a test corpus of human dialogue utterances to see how well a selection approach could approximate these, given a training corpus of utterances in that domain. We make the following assumptions to allow automatic evaluation of appropriateness of a selected utterance.

- Actual human dialogues represent a gold-standard for computer systems to emulate; i.e. choosing an actual utterance in the correct place is the best possible result.
- Other utterances can be evaluated as to how close they come to the original utterance, using a similarity metric.

Then the maximum performance for the *selection* approach will be the proportion of test utterances for which an exact or approximate match exists in the corresponding training corpus. We look at exact matches as well as utterances having their similarity score above a threshold. We investigate the effect of the size of training corpora, which lets us know how much data we might need to achieve a certain level of performance.

Given a set of utterances extracted from a training corpus U_{train} and a similarity function f , we calculate the score for a test utterance u_{test} as,

$$maxsim_f(u_{test}) = \max_k f(u_k, u_{test}) \quad \forall u_k \in U_{train} \quad (4.1)$$

The higher the expected value of $maxsim_f(u_{test})$ the more likely it is that an utterance similar to u_{test} has been seen before and this domain will be more amenable to the *selection* approach.

There are several choices for the utterance similarity function f . Ideally such a function would take meaning and context into account along with the surface text. But these aspects are harder to automate, so for our initial experiments we look at several metrics working at the surface text level alone, as described below.

The **Exact** measure returns 1 if the utterances are exactly the same and 0 otherwise. **1-WER**, a similarity measure related to word error rate, is defined as $\min(0, 1 - levenshtein(u_{test}, u_k)/length(u_{test}))$. **METEOR** (Lavie and Denkowski, 2009), one of the automatic evaluation metrics used in machine translation, is a good candidate for f . METEOR finds optimal word-to-word alignment between test and reference strings based on several modules that match exact words, stemmed words and synonyms. METEOR is a tunable metric and for our analysis we used the default parameters tuned for the Adequacy & Fluency task.

All of the previous measures take into account the word ordering of test and reference strings. In contrast, document similarity measures used in information retrieval generally follow the *bag of words* assumption, where a string is converted to a set of tokens. Here we also considered **Cosine** and **Dice** coefficients using the standard boolean model. In our experiments, the surface text was normalized and all punctuation was removed.

We conducted our experiments on several different dialogue genres and domains. SGT Blackwell (Leuski et al., 2006) and SGT Star (Artstein et al., 2008) are simple question-answering characters who can answers questions about themselves, their technology, the U.S. Army and careers in the army. Amani, presented in chapter 3, is an advanced question answering virtual human used for training soldiers skills for tactical questioning. The SASO-ST system, presented in section 4.1, is a two-party negotiation training prototype where a human trainee negotiates with a virtual character. The SASO-EN system (Traum et al., 2008b) is a multi-party extension of the SASO-ST system in which two virtual characters negotiate with a human trainee. The Radiobots system (Roque et al., 2006b) is a training prototype that responds to military calls for artillery fire. IOTA (Roque et al., 2010) is an extension of the Radiobots system where corpus consists of training sessions between a human trainee and a human instructor on a variety of missions. Other corpora involved dialogues between two people playing specific roles in planning, scheduling problems for railroad transportation, the Trains-93 corpus (Heeman and Allen, 1994) and for emergency services, the Monroe corpus (Stent, 2000). The Switchboard corpus (Godfrey et al., 1992) consists of telephone conversations between two people, based on provided topics. These corpora differ along a number of dimensions such as the size of the corpus, dialogue genre (question-answering, transactions, negotiations or conversational), dialogue domains (artillery calls, moving and scheduling resources or information about the U.S. Army and careers in it) and motivation of the participants (exploring a new technology – SGT Blackwell, presenting a demo – SGT Star, undergoing training – Amani, IOTA or simply for collecting the corpus – Switchboard, Trains-93, Monroe). While the set of corpora we include does not cover all points in these dimensions, it does present an interesting range.

Table 4.1 reports the mean of $maxsim_f$ scores for several domains. These can be interpreted as the expectation of $maxsim_f$ score for a new test utterance and indicates how amenable a particular domain is to the *selection* approach. All the similarity measures we used are surface text based and the choice of similarity function f does not have a large effect on $maxsim_f$. The correlation between $maxsim_f$ for different choices of f (except Exact match) is very high (Pearson’s $r > 0.94$). The table also shows the percentage of utterances that had a $maxsim_{Meteor}$ score above a certain threshold.

Domains like SGT Star, Blackwell, Radiobots and SASO are better suited for *selection* approaches. Whereas, domains like Trains-93, Monroe, Switchboard and Amani are not best suited for *selection* approaches, at least with the amount of data we have available. The IOTA domain falls somewhere in between these two domain classes. For more detailed information about the results and description of various domains used please refer to [Gandhe and Traum \(2010\)](#). The important finding from this analysis is that some but not all domains are suitable for *selection* approach.

Of special interest to us is the SASO-ST domain – the test-bed for our experiments with unsupervised dialogue models. Here we report the analysis results for each speaker role. It can be seen that the role of the doctor from SASO-ST domain is amenable to the *selection* approach where about 63% of test utterances have been seen before in the training corpus. But the *selection* approach is not suitable for the role of the captain given the available training data.

Domain	Train		Test		$mean(maxsim_f)$				% of utterances		
	# utt	words	# utt	words	MET - EOR	1-WER	Dice	Cosine	Exact	METEOR ≥ 0.9	METEOR ≥ 0.8
Blackwell	17755	84.7k	2500	12.0k	0.913	0.878	0.917	0.921	69.6	75.8	82.1
Radiobots	995	6.8k	155	1.2k	0.905	0.864	0.920	0.924	53.6	67.7	83.2
SGT Star	2974	16.6k	400	2.2k	0.897	0.860	0.906	0.911	65.0	70.5	78.0
SASO-EN	3602	23.3k	510	3.6k	0.821	0.742	0.830	0.837	38.4	48.6	62.6
IOTA	4935	50.4k	650	5.6k	0.768	0.697	0.800	0.808	36.2	42.8	51.4
Trains 93	5554	47.2k	745	6.0k	0.729	0.633	0.758	0.769	34.5	36.9	42.8
SWBD ¹	19741	138.2k	3173	21.5k	0.716	0.628	0.736	0.753	35.8	37.9	44.2
Amani	1455	15.8k	182	1.9k	0.675	0.562	0.694	0.706	18.7	25.8	30.8
Monroe	5765	43.0k	917	8.8k	0.594	0.491	0.639	0.658	22.3	23.6	26.1
SASO-ST											
– Captain	597	16.8k	88	2.4k	0.453	0.319	0.535	0.551	5.7	6.8	12.5
– Doctor	600	9.9k	89	1.2k	0.791	0.742	0.819	0.826	62.9	66.3	67.4

Table 4.1: Corpus details for different domains, expected value of $maxsim_f$ scores along with percentage of utterances with exact or approximate match.

Figure 4.5 shows the effect of training data size on the expected value of $maxsim_{Meteor}$ score for different domains. Radiobots domain shows very high scores even for small amounts of training data. SGT Star and SGT Blackwell also converge fairly early. Switchboard, on the other hand, does not achieve very high scores even with a large number of utterances. For all domains, with around 2500 training utterances $maxsim_{Meteor}$ reaches 90% of its maximum possible value for the training set. Figure 4.6 shows that SASO-ST domain converges even with small amounts of available data.

4.5 Selection Criterion

We need a suitable criterion for selecting an appropriate response utterance for a given dialogue context. One approach is to make a binary decision regarding whether a response utterance is appropriate or not

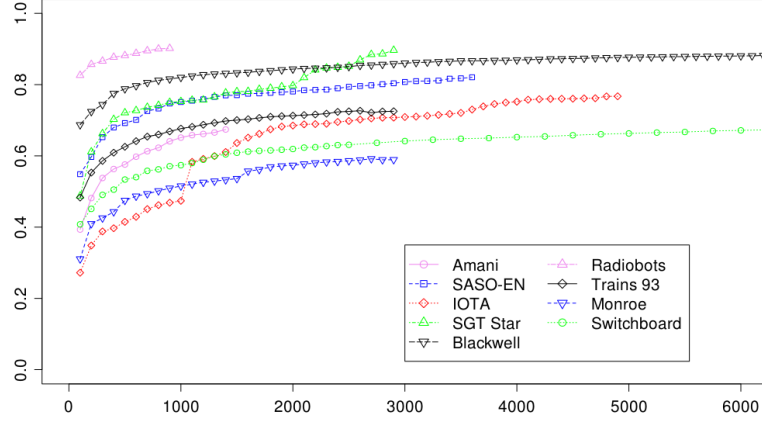


Figure 4.5: Expected value of $maxsim_{Meteor}$ vs # utterances in the training data for different domains.

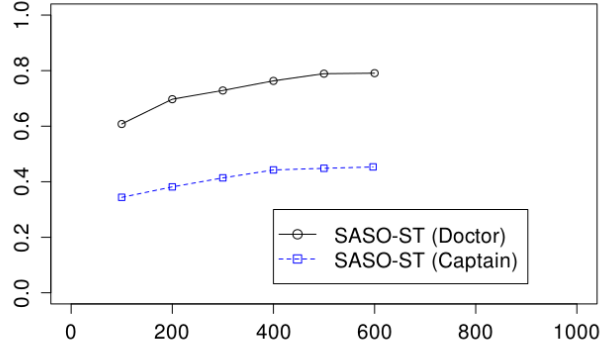


Figure 4.6: Expected value of $maxsim_{Meteor}$ vs # utterances in the training data for SASO-ST domain.

for a given dialogue context and then randomly pick one from the set of responses that were judged appropriate. Another approach is to rank all possible response utterances using a suitable ranking function and then pick the top-ranked response. Yet another approach would be to build a posterior probability distribution for a response utterance given the dialogue context and use this distribution to sample from existing set of possible responses.

If we assume that the human-human dialogues observed during training are optimal then one suitable criterion would be to imitate these dialogues. We can build a model for predicting the next utterance given the context based on the training data and use it to choose an utterance given an unseen context during testing. This decision criterion states that **the most appropriate response is the most probable response according to a model trained on human-human dialogues**. Here the task of dialogue is being modeled as that of the prediction of a sequence of tokens similar to language modeling.

Let a dialogue be represented as a sequence of utterances $\langle u_1, u_2, \dots, u_{t-1}, u_t, \dots, u_T \rangle$, where T is the length of the dialogue. We build a model $P(u_t | context_t)$ that predicts the next utterance, u_t given

the $context_t$ which represents the dialogue fragment $\langle u_1, u_2, \dots, u_{t-1} \rangle$. P is the posterior probability distribution estimated from in-domain human-human dialogue corpus. It is not straightforward to estimate this probability distribution.

If we are to use the ranking approach the most appropriate response will be selected as,

$$u_t = \underset{i}{\operatorname{argmax}} \quad P(u_i | context_t) \quad \forall u_i \in U_{possible} \quad (4.2)$$

where $U_{possible}$ is a set of all possible response utterances.

For the ranking approach, exact probabilities are not required and only the top-ranked response needs to be identified. In the dialogue models we implement, we use the ranking approach as our selection criterion. We approximate $P(u_i | context_t)$ by a ranking function $R(u_i, context_t)$. This ranking function is trained from the training corpus with a goal of imitating the training corpus. Each dialogue model uses a different ranking function based on the available resources from the training corpus. We further assume $U_{possible} = U_{train}$, the set of utterances observed in training corpus. For every utterance $u_i \in U_{train}$, we also have observed the corresponding context $context_i$ in the training corpus.

Other selection criteria are also possible. There have been efforts that model dialogue management as a sequential decision making process such as MDP (Levin et al., 1997) or POMDP (Williams and Young, 2007). In such frameworks the selection criterion is to choose the response utterance that maximizes the expected utility. These frameworks require that a reward function be specified for every action from every state. Designing such a reward function remains a challenging process which requires a lot of manual tweaking. Recently there have been efforts where the reward function is based on how closely the dialogue model fits the observed data (Minami et al., 2011). All such sequential decision making approaches require a small state space to remain tractable and hence need to be operated at the dialogue act level. They require considerable expertise in designing the state space and require annotations for dialogue acts. Since our models need to work primarily at surface text level we will be using the selection criterion as specified in equation 4.2.

4.6 Unsupervised Dialogue Models

We have implemented six types of unsupervised dialogue models that employ the *selection* approach for formulating a response utterance and use a ranking function trained from training corpus as the selection criterion. The ranking function used by each dialogue model uses a different set of resources. Figure 4.7 shows a schematic representation of these models along with the set of resources being used by each model. The figure also shows the relationships between these models. The arrows point from a less informative model to a more informative model and the annotations on these arrows indicate the additional information used.

The dialogue model at the bottom, *Random* (Section 4.6.1), is a baseline model that uses an uninformative ranking function. *Segmented Random* model (Section 4.6.4) is another baseline model that uses an additional resource of topic. *Nearest Context* model (Section 4.6.2) is based on information retrieval techniques and uses the context information for ranking. *Segmented Nearest Context* model (Section 4.6.3) uses a ranking function based on both the topic and the context information. The context information represents local aspect of dialogue coherence while the topic represents a global aspect. Both *Nearest Context* and *Segmented Nearest Context* models do not use the actual content of the response utterance u_i . For all the models except *Random*, the ranking function is defined only over the response utterances $u_i \in U_{train}$ because the ranking function requires access to $context_i$.

The next two advanced models, *Cross-lingual Relevance Model* and *Perceptron*, take into account the actual contents of the response utterance u_i along with the contents of the contexts ($context_i$ &

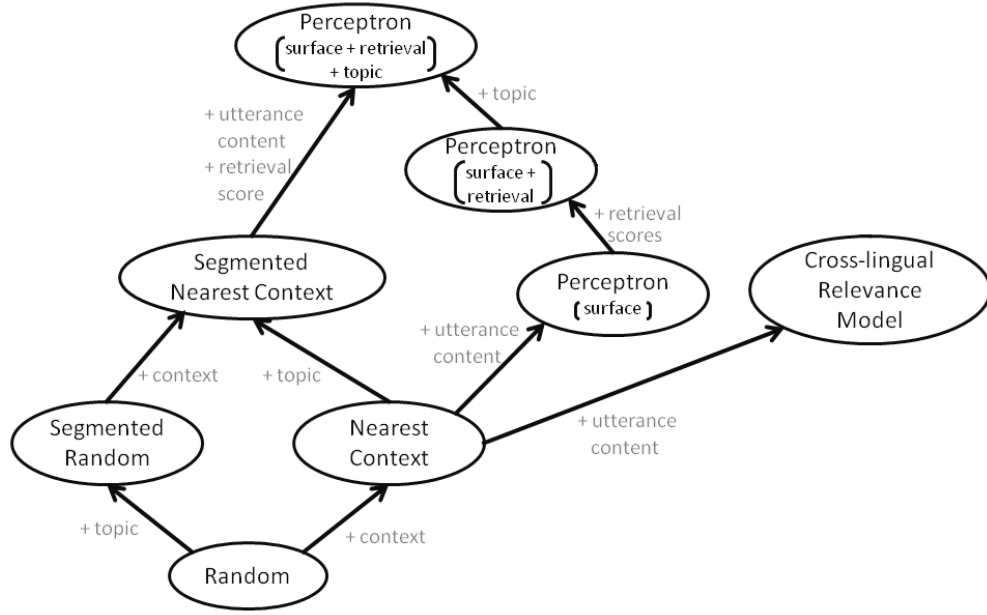


Figure 4.7: A schematic representation of implemented unsupervised dialogue models and the relationships between the information used by their ranking functions.

$context_t$). These two models are not restricted to ranking only the utterances from U_{train} but can rank an arbitrary set of utterances including utterances whose contexts are not known. Section 4.6.5 presents a model based on cross-lingual relevance models (Lavrenko et al., 2002). These were first used for dialogue modeling in simple question-answering dialogue by Leuski et al. (2006). Section 4.6.6 presents a model based on averaged perceptron (Collins, 2002) which has been used for various NLP tasks such as part of speech tagging, parsing, language modeling. Here we use it for unsupervised dialogue modeling.

4.6.1 Random

This model provides a zero baseline and does not capture global or local coherence. A set of utterances with the doctor as the speaker is compiled from the corpus. The model just replies to any utterance from the trainee captain with a randomly selected utterance from this list. There are 508 unique doctor utterances to randomly choose from. Note that, the random response is always relevant within the scenario, if not coherent with respect to the dialogue context.

If N is the total number of utterances that can be a system's response then,

$$P(u_i|context_t) = P(u_i) = \frac{1}{N} \quad (4.3)$$

Here rather than estimating the probabilities for $P(u_i)$ from the training corpus, we assume a uniform distribution. This allows for some variety in system's responses rather than repeating the most frequent

utterance (e.g., “hello”). Since all the system’s response utterances are within the domain of the application, some responses may seem appropriate. An example interaction in the SASO-ST domain using this model in dynamic context setting is shown in Figure 4.8.

Captain	hello
Doctor	so how are you going to do that
Captain	hello doctor i am captain kirk
Doctor	uh i have i have patients from both sides from ah there have been injured from american forces as well as other locals

Figure 4.8: Example interaction for *Random* in dynamic context setting.

4.6.2 Nearest Context

This model tries to capture the local context. Here the context is approximated by the previous n turns ($n = 2$). For this model we use a simple ranking function for relevance, R , that is defined over the set of utterances in the training data. From the training data we extract a set of pairs $\langle u_i, context_i \rangle$, where the utterance u_i has been seen in the dialogue with context $context_i$. When it’s time to choose the response utterance for the given $context_t$, we find a $context_i$ which is most similar to $context_t$ of the current dialogue. The utterance u_i associated with $context_i$ will be the system’s reply. The selection criterion becomes,

$$P(u_i|context_t) \approx R(u_i, context_t) = Sim(context_i, context_t) \quad \forall u_i \in U_{train} \quad (4.4)$$

Here Sim is a similarity function for comparing contexts. Following the vector-space model (Salton and McGill, 1983), the *context* is represented by *tf-idf* weighted vector of the words that occurred in previous n turns. The features used are stemmed unigrams augmented with speaker and distance in time in units of turns. The latest turn is at a distance of 0, the previous at 1 and so on. For the systems to be more reactive to the latest input, we weigh these *tf-idf* scores depending on how far back in the history the utterance is.

Let W_i^j be the weight assigned for unigram w_i which appears j turns ago. Then W_i^j is given by,

$$W_i^j = TF(w_i) \times IDF(w_i) \times H(j) \quad (4.5)$$

$$TF(w_i) = 1 + \log(\#w_i) \quad (4.6a)$$

where $\#w_i$ is the number of times w_i appears in the utterance

$$IDF(w_i) = \log\left(\frac{N}{df_i}\right) \quad (4.6b)$$

where N is the total number of utterances and df_i is the number of utterances containing w_i

$$H(j) = \exp\left(\frac{-j^2}{2}\right) \quad (4.6c)$$

		:
u_{t-2} :	Doctor	what do you want i have patients waiting for me
u_{t-1} :	Captain	I have orders to assist you in moving this clinic
<hr/>		
	<captain>_0_orders	6.24610677
	<captain>_0_assist	4.16666522
	<captain>_0_moving	2.98801023
	< doctor>_1_waiting	2.45577118
	<captain>_0_clinic	2.41746537
	<captain>_0_in	2.18566375
	< doctor>_1_want	1.86086817
	<captain>_0_this	1.73277930
	< doctor>_1_me	1.66771622
	<captain>_0_have	1.66113929
	< doctor>_1_for	1.32567208
	< doctor>_1_what	1.21161264
	< doctor>_1_do	1.18616006
	<captain>_0_i	1.08705147
	<captain>_0_to	1.03117101
	< doctor>_1_patients	1.02003814
	< doctor>_1_have	1.00753191
	< doctor>_1_i	0.65933004
	<captain>_0_you	0.56935296
	< doctor>_1_you	0.34533003

Figure 4.9: Feature vector representing the context $\langle u_{t-2}, u_{t-1} \rangle$ of previous $n = 2$ turns.

Figure 4.9 shows an example of the feature vector used to represent the *context*. The context similarity function used is,

$$Sim(context_i, context_t) = Cosine(\overline{context_i}, \overline{context_t}) \quad (4.7)$$

where $\overline{context_i}$ and $\overline{context_t}$ are the feature vectors for $context_i$ and $context_t$.

The model implements a nearest neighbor classifier for predicting the most appropriate utterance based on context similarity. Capturing the local context alone makes the resulting virtual human act in a reactive way. Figure 4.10 shows an example interaction with this model in dynamic context setting.

When this model is being used in a dynamic context setting during a conversation with an user, we modify the dialogue manager slightly in order to avoid repetitions in the dialogue. If the predicted best response u_t , is same as the previous system response u_{t-2} , then we choose the next best response utterance.

4.6.3 Segmented Nearest Context

The *Nearest Context* model, presented in previous section tries to capture local coherence alone. It captures local features by approximating dialogue context with the previous two turns. There are cases where dialogue context cannot be faithfully represented using only the previous two turns. E.g., during

Captain	hello doctor
Doctor	hello
Captain	i am captain kirk nice to meet you
Doctor	i'm uh doctor perez uh i'm very busy right now i i only have uh just a few minutes to uh
Captain	i want to talk to you about an important issue
Doctor	uh captain i i appreciate you have given us a lot of very good information to uh which roads are safe and where the landmines are and i need you uh i i cannot move this location though because of all these patients they're they're too critical right now i'm working on a on a on a young girl with amoebic dysentery and and she she requires my attention at all times i there is no way i i these people are in no shape to to

Figure 4.10: Example interaction for *Nearest Context* in dynamic context setting.

a negotiation, a response to a proposal may depend on what other proposals have been agreed to earlier in the dialogue. Such agreements, social commitments may not be evident in the local features extracted from previous few turns. E.g., In the SASO-ST domain the doctor would not accept the proposal to move the clinic unless he has been promised help with the transportation.

Besides capturing only local features, the *Nearest Context* model sometimes produces inappropriate responses due to a simplistic context matching function and the non-availability of a large enough training corpus. E.g., In the last utterance in Figure 4.10 doctor saying “I cannot move” seems unwarranted given the dialogue context.

The *Segmented Nearest Context* model tries to remedy this by also capturing dialogue features at a global level that are required to track such long distance dependencies. In order to track these additional global features of the context, we augment the $context_t$ representation with the information state IS_t . We use the same selection criterion as *Nearest Context* model but with augmented context representation,

$$P(u_i|context_t) \approx R(u_i, context_t) = Sim_{seg}(context_i, context_t) \quad \forall u_i \in U_{train} \quad (4.8)$$

where $context_t = \langle u_{t-1}, u_{t-2}, IS_t \rangle$ and $IS_t = update(IS_{t-1}, u_{t-1})$. In this model, the response utterance is conditioned on the previous two utterances as well as information state (IS_t) at the current time instant. This information state can contain any of the dialogue features that would be found in traditional information state based dialogue system (e.g., current topic of conversation, social commitments of conversational participants, proposals agreed/disagreed/discussed). The *update* function is responsible for incrementally updating the information state. In a traditional information state based dialogue model, this *update* function would require a dialogue act interpretation of the incoming utterance. Moreover the actual update function is generally rule-based and these rules are manually authored. Here in order to allow rapid prototyping we restrict the *update* function to operate at the surface text level.

In the SASO-ST domain with the *Nearest Context* model, we observed that most of the inappropriate responses could be attributed to unwarranted presuppositions. Stalnaker gives a pragmatic account of presupposition and defines it as,

“A speaker presupposes that P at a given moment in a conversation just in case he is disposed to act, in his linguistic behavior, as if he takes the truth of P for granted, and as if he assumes that his audience recognizes that he is doing so.” (Stalnaker, 1973)

When a response makes unwarranted presuppositions, it leads to failure in dialogue coherence. One such problem is illustrated by the last utterance in Figure 4.10. The doctor saying “I cannot move the clinic” presupposes that “doctor may need to move the clinic”. It makes sense only if both the dialogue participants believe that the doctor moving the clinic is under discussion. In this case, the context of the interaction does not support such an assumption. The doctor should not already believe that he may need to move the clinic. This unwarranted presupposition results in a lack of coherence.

We need to track global features of dialogue that allow us to avoid such unwarranted presuppositions. Additionally we need to work primarily at the surface text level. For these reasons, we annotate each dialogue with an information state variable that keeps track of whether several key concepts were introduced earlier in the dialogue or not. Tracking the mentions of these concepts effectively splits the dialogue into various segments each with a specific topic signature. In our model this topic signature is used as the information state. These key concepts or topics are picked by an expert who understands the domain well. Figure 4.11 illustrates the list of key concepts for the SASO-ST domain. It identifies the concepts along with the words that are used to represent them. The *update* function tracking information state is as simple as keyword spotting. Figure 4.12 shows an example interaction with this dialogue model in dynamic context setting. It also shows the topic signature at every turn.

[move]	move , relocate
[supplies]	antibiotics, supplies, medicines, plasma
[transportation]	vehicle, trucks, transportation

Figure 4.11: List of key concepts along with the representative unigrams compiled for SASO-ST dialogues.

		[move, supplies, transcription]
Captain	hello doctor	[false,false,false]
Doctor	hello	[false,false,false]
Captain	i am captain kirk nice to meet you	[false,false,false]
Doctor	i'm uh doctor perez uh i'm very busy right now i i only have uh just a few minutes to uh	[false,false,false]
Captain	i want to talk to you about an important issue	[false,false,false]
Doctor	ok	[false,false,false]
Captain	I would like to request you to move this clinic to another location	[true,false,false]

Figure 4.12: Example interaction for *Segmented Nearest Context* in dynamic context setting. The third column shows the information state consisting of the topic signature.

The context similarity function used for this model takes into account the information state. There are different possibilities for how to combine information from local dialogue context (previous 2 turns) and information state variables. In our case IS_t consists of a single variable, topic signature indicating mentions of key concepts. We use a simple similarity function,

$$Sim_{seg}(context_i, context_t) = \begin{cases} Cosine(\overline{context_i}, \overline{context_t}) & \text{if } IS_i = IS_t \\ 0 & \text{otherwise} \end{cases} \quad (4.9)$$

where $\overline{context_t}$ is a feature vector representing $\langle u_{t-1}, u_{t-2} \rangle$ in the same fashion as *Nearest Context* model.

4.6.4 Segmented Random

The *Segmented Nearest Context* model tries to achieve local appropriateness by finding the most similar context and also the global appropriateness by keeping track of the segments. To understand which of the two factors makes more significant impact we implemented the fourth model. This *Segmented Random* model tries to capture only the global features and ignores local context. It keeps track of the topic signature of the context but picks up one of the utterance randomly with that signature. The next utterance is only dependent on the current information state. The corresponding selection criterion is,

$$P(u_i | context_t) = P(u_i | IS_t) = \begin{cases} 1/N_i & \text{if } IS_i = IS_t \\ 0 & \text{otherwise} \end{cases} \quad \forall u_i \in U_{train} \quad (4.10)$$

Where N_i is the number of utterances from the training data whose information state (i.e. topic signature), IS_i , matches the current information state. Figure 4.13 shows an example interaction with the *Segmented Random* model.

Captain	hello
Doctor	yes
Captain	i am captain kirk nice to meet you
Doctor	you are the threat i need protection from you
Captain	no we are here to protect you and your clinic
Doctor	are you injured

Figure 4.13: Example interaction for *Segmented Random* model in dynamic context setting.

4.6.5 Cross-lingual Relevance Model

Leuski et al. (2006) presented an approach based on cross-language information retrieval for simple question-answering virtual characters. This model works at the surface text level and can rank a set of possible responses given an unseen question. This model requires a set of questions, a set of answers and a set of links between questions and answers. Each link connects a question with an appropriate answer. These links are generally manually authored. Here, we use this model to map the input dialogue context $context_t$ to one of the response utterances from the training set U_{train} . The model views this mapping as a cross-language information retrieval task where the $context_t$ forms the *query* and the utterances are *documents*. Similar to the *Nearest Context* model, the training data comprises of a set of pairs $\langle u_i, context_i \rangle$ henceforth referred to as the *parallel corpus*.

The model performs the task of cross-language information retrieval based on *Cross-lingual Relevance Model* proposed by Lavrenko et al. (2002). Given an unseen query (Q), this approach first builds a language model (P_Q) for how likely it is that a given word will belong to the relevant *document*. This language model is called the *relevance model* and is estimated based on the *parallel corpus* (\mathcal{L}). The relevance model is then used to rank the set of all *documents* (\mathbb{A}). The ranking function is based on the Kullback-Leibler (K-L) divergence between the relevance model (P_Q) and the language model estimated from the individual *documents* themselves (P_A). A lower K-L divergence value indicates a more relevant

document. Hence, the score received by a *document* (A) in response to a query (Q) will be $-D(P_Q||P_A)$. The selection criterion used by this model is,

$$R(u_i, context_t) = R(A, Q) = -D(P_Q||P_A) = - \sum_{a \in V_{\mathbb{A}}} P_Q(a) \log \frac{P_Q(a)}{P_A(a)} \quad (4.11)$$

where $V_{\mathbb{A}}$ is the vocabulary defined over all *documents*. P_A is the language model estimated from the document A using maximum likelihood estimate with Jelinek-Mercer smoothing (Jelinek and Mercer, 1980) and is defined as,

$$P_A(a) = \lambda \frac{df_A(a)}{|A|} + (1 - \lambda) \frac{cf_{\mathbb{A}}(a)}{|\mathbb{A}|} \quad (4.12)$$

where $df_A(a)$ is the number of times the word a is seen in the document A , $|A|$ is the total number of words in the document A , $cf_{\mathbb{A}}(a)$ is the collection frequency of the word a , $|\mathbb{A}|$ is the total number of words in the collection \mathbb{A} and λ is a tunable smoothing parameter. The relevance model P_Q for a query $Q = \langle q_1, q_2, \dots, q_n \rangle$ is estimated as,

$$P_Q(a) = P(a|Q) = \frac{p(a, q_1, q_2, \dots, q_n)}{p(q_1, q_2, \dots, q_n)} \quad (4.13)$$

In equation 4.13, both the numerator and the denominator are joint probability distributions which are to be estimated based on the *parallel corpus*. The parallel corpus (\mathcal{L}) is a set of links in the form $\langle A_l, Q_l \rangle$. The required joint distributions are estimated using kernel-based density estimation (Lavrenko, 2004). The specific kernel used here is the delta-kernel.

$$P(Q) = P(q_1, q_2, \dots, q_n) = \frac{1}{|\mathcal{L}|} \sum_{\langle A_l, Q_l \rangle \in \mathcal{L}} P_{Q_l}(Q) \quad (4.14)$$

$$P(a, Q) = P(a, q_1, q_2, \dots, q_n) = \frac{1}{|\mathcal{L}|} \sum_{\langle A_l, Q_l \rangle \in \mathcal{L}} P_{A_l}(a) P_{Q_l}(Q) \quad (4.15)$$

$$P_{Q_l}(Q) = P_{Q_l}(q_1, q_2, \dots, q_n) = \prod_{i=1}^n P_{Q_l}(q_i) \quad (4.16)$$

$|\mathcal{L}|$ is the number of links in the parallel corpus. The approach used here to estimate the joint probability distributions is a non-parametric approach. In order to compute the overall probability ($P(Q)$) of a possibly unseen query (Q), we first go through all examples of queries from our training data. At each example (Q_l), we compute the probability $P_{Q_l}(Q)$, estimated by looking at this single observation. Intuitively $P_{Q_l}(Q)$ indicates how similar the unseen query (Q) is to a seen example (Q_l). To get the overall probability $P(Q)$, we simply average the individual $P_{Q_l}(Q)$ values giving equal weight to each example. This inference runs in linear-time $O(|\mathcal{L}|)$.

The formulation presented so far assumes that the words for queries (q_i) are drawn from a single language and this language is different from the one for the words in the documents (a_i). This framework has been extended to cases where queries and documents are composed of different heterogeneous *fields* (Leuski and Traum, 2010). If a query has K fields such that $Q = \langle q_1^1, \dots, q_{n_1}^1, \dots, q_1^k, \dots, q_{n_k}^k, \dots, q_1^K, \dots, q_{n_K}^K \rangle$ then equation 4.16 becomes,

$$P_{Q_l}(Q) = P_{Q_l}(q_1^1, \dots, q_{n_1}^1, \dots, q_1^k, \dots, q_{n_k}^k, \dots, q_1^K, \dots, q_{n_K}^K) = \prod_{k=1}^K \left(\prod_{i=1}^{n_k} P_{Q_l^k}(q_i^k) \right)^{\beta_k} \quad (4.17)$$

where β_k are tunable parameters indicating the relative weight of each field with a constraint, $\sum \beta_k = 1$. The overall model presented here is non-parametric but it still has a few hyper-parameters. Each field in queries and documents has a smoothing parameter λ and a weight parameter β . These handful of parameters can be set manually or tuned on a separate development set.

In our case the *queries* are contexts representing the previous two turns which can be thought of as two different fields. The corresponding utterances are *documents*. In fact, contexts can include additional fields for information state components such as topic or dialogue act. DeVault et al. (2011a) used this model in the task of choosing an appropriate response dialogue act for a given dialogue context. The dialogue context was represented using previous user and system dialogue acts along with the surface text. But, every additional field adds parameters and tuning them requires a substantial development set and an evaluation metric. Such an evaluation metric should be automatic and a reliable estimate of the model's performance on previously unseen contexts. We use this model for the task of choosing an appropriate response utterance for a given dialogue context and there is no straightforward reliable evaluation metric. For our model we used the implementation that is available as a part of NPCEditor (Leuski and Traum, 2011). Due to absence of a reliable automatic evaluation and non-availability of a substantial development corpus we manually set the parameters to reasonable values ($\lambda = 0.9$ and $\beta_1 = \beta_2 = 0.5$).

	Original Human-Human Dialogue	Model response
Captain	hello doctor perez	
Doctor	hello	hello what was your name captain dezois very nice to meet you i am sorry but i am very busy today so i only have a limited amount of time what can i help you with
Captain	i am captain xx	
Doctor	so do you need help	ok what do you want i have patients waiting for me
Captain	yes i have a very urgent matter to discuss with you	
Doctor	are you injured	pl please captain i i am sorry to cut you off but i really must uh go i was in the middle of examining a patient

Figure 4.14: Example interaction for *Cross-lingual Relevance Model* model in static context setting. The second column shows the original human-human dialogue and the third column shows the *Cross-lingual Relevance Model* model's response for the corresponding system turn.

4.6.6 Perceptron

The selection criterion as shown in equation 4.2 requires an estimate for $P(u_i|context_t)$. Standard classification approaches such as maximum entropy (Berger et al., 1996) could provide estimates for such a posterior. The task of selecting the most appropriate response can be viewed as a multi-class classification. But there are a couple of issues with a simple classification approach. First, since we operate at the surface text level, each unique response utterance will be labeled as a separate class. The

number of classes is the number of unique utterances in the set U_{train} , which is relatively large. As the training data grows, the number of classes will increase. Second, there are very few examples (on average a single example) per class. We need a classifier that can overcome these issues.

The perceptron algorithm and its variants – voted perceptron and averaged perceptron are well known classification models (Freund and Schapire, 1999). They have been extended for use in various natural language processing tasks such as part-of-speech tagging (Collins, 2002), parsing (Collins, 2004) and discriminative language modeling (Roark et al., 2007). Here we use the averaged perceptron model for mapping from dialogue context to an appropriate response utterance.

Collins (2002) outlines the following four components of a perceptron model:

- The training data. In our case it is the parallel corpus $\{\dots, \langle u_i, context_i \rangle, \dots\}$
- A function $GEN(context)$ that enumerates a set of all possible outputs (response utterances) for any possible input (dialogue context)
- A feature extraction function $\Phi : \langle u, context \rangle \rightarrow \mathbb{R}^d$ that is defined over all possible pairings of response utterances and dialogue contexts. d is the total number of possible features.
- A parameter vector $\bar{\alpha} \in \mathbb{R}^d$

Using such a perceptron model, the most appropriate response utterance (u_t) for the given dialogue context ($context_t$) is given by,

$$u_t = F(context_t) = \underset{u \in GEN(context)}{\operatorname{argmax}} \Phi(u, context_t) \cdot \bar{\alpha} \quad (4.18)$$

Algorithm 4.1 Perceptron Training Algorithm

```

Initialize:  $t \leftarrow 0$ ;  $\bar{\alpha}_0 \leftarrow 0$ 
for  $iter = 1$  to  $MAX\_ITER$  do
  for  $i = 1$  to  $N$  do
    set  $r_i \leftarrow F(context_i) = \operatorname{argmax}_{u \in GEN(context_i)} \Phi(u, context_i) \cdot \bar{\alpha}_t$ 
    if  $r_i \neq u_i$  then
      /* Prediction error happened, Update the  $\bar{\alpha}$  */
       $\bar{\alpha}_{t+1} \leftarrow \bar{\alpha}_t + \Phi(u_i, context_i) - \Phi(r_i, context_i)$ 
    else
       $\bar{\alpha}_{t+1} \leftarrow \bar{\alpha}_t$ 
    end if
     $t \leftarrow t + 1$ 
  end for
end for
return  $\bar{\alpha} \leftarrow (\sum_t \bar{\alpha}_t) / (MAX\_ITER \times N)$ 

```

The parameter vector $\bar{\alpha}$ is trained using the training algorithm described in Algorithm 4.1. The algorithm goes through the training data one instance at a time. For every training instance, it computes the best response utterance (r_i) for the context based on its current estimate of the parameter vector $\bar{\alpha}_t$. The algorithm changes the parameter vector only if it makes an error ($r_i \neq u_i$). The update drives the parameter vector away from the error (r_i) and towards the correct output (u_i). The final parameter vector $\bar{\alpha}$ is an average of all the intermediate $\bar{\alpha}_t$ values. This averaging parameter vectors avoid overfitting.

The feature extraction function Φ can list any arbitrary features from the pair $\langle u, context \rangle$. The context can include information state annotations (IS_t) along with the surface text corresponding to the previous $n = 2$ turns. The features could also include scores computed from other models. Let's consider some possible features that can be extracted from a pair $\langle u_i, context_j \rangle$:

Surface text based features (Φ_S) are the features extracted from the surface text of the previous utterances in the dialogue context ($context_j$) and the response utterance (u_i). Let $context_j = \langle u_{j(-1)}, u_{j(-2)} \rangle$ represent the previous two turns in the context then,
 $\Phi_S(\langle u_i, context_j \rangle) = \Phi_{S(-1)}(u_i, u_{j(-1)}) \cup \Phi_{S(-2)}(u_i, u_{j(-2)})$ where,
 $\Phi_{S(d)}(u_x, u_y) = \{cross_term(d, w_x, w_y) : w_x \in u_x \wedge w_y \in u_y \wedge \langle w_x, w_y \rangle \in Selected_cross(d)\} \cup$
 $\{common_term(d, w) : w \in u_x \wedge w \in u_y \wedge w \in Selected_common(d)\} \cup$
 $\{common_term_count(d)\} \cup$
 $\{unique_common_term_count(d)\}$

$\Phi_{S(d)}$ extracts surface text features from two utterances – a response utterance (u_x) and an utterance (u_y) from the context that is at a distance (d) in time. There are four types of features we can extract.

- $cross_term(d, w_x, w_y)$ features indicate the number of times the word w_x appears in the utterance u_x and the word w_y appears in the utterance u_y . The total possible number of such cross features is very large ($O(|V|^2)$), where $|V|$ is the utterance vocabulary size. In order to keep the training tractable and avoid overfitting, we select a small subset of cross_features ($Selected_cross(d)$) from all possible features.
- $common_term(d, w)$ features indicate the number of times a word w appears in both the utterances. Again, the total number of possible features is $O(|V|)$ and we select a small subset of words ($Selected_common(d)$) from the vocabulary.
- $common_term_count(d)$ feature indicates the number of words common in these two utterances.
- $unique_common_term_count(d)$ feature indicates the number of unique words common in these two utterances.

In this model, we perform feature selection by selecting the subsets $Selected_cross(d)$ and $Selected_common(d)$. The training algorithm requires evaluating the feature extraction (Φ_S) function for all possible pairings of response utterances and contexts. One simple feature selection criterion is to allow the features only appearing in *true pairings* of response utterance and context (i.e. features from $\Phi_S(\langle u_i, context_j \rangle) \forall i = j$). The subset $Selected_common(d)$ for $common_term$ features is selected by extracting features from only such *true pairings*.

For selecting $cross_term(d, w_x, w_y)$ features we use only *true pairings* but we need to reduce this subset even further. We impose additional constraints based on the collection frequency of lexical events such as, $cf(w_x) > threshold_x$, $cf(w_y) > threshold_y$, $cf(\langle w_x, w_y \rangle) > threshold_{xy}$. Further reduction in size of the selected subset of $cross_term$ features is achieved by ranking the features using a suitable ranking function and choosing the top n features. In this model, we rank the $cross_term$ features based on pointwise mutual-information.

$$pmi(\langle w_x, w_y \rangle) = \log \frac{p(\langle w_x, w_y \rangle)}{p(w_x)p(w_y)} = \log \frac{\left(\frac{\# \langle w_x, w_y \rangle}{\# \langle \cdot, \cdot \rangle} \right)}{\left(\frac{\# \langle w_x, \cdot \rangle}{\# \langle \cdot, \cdot \rangle} \right) \cdot \left(\frac{\# \langle \cdot, w_y \rangle}{\# \langle \cdot, \cdot \rangle} \right)}$$

The ranking scores computed from any of the previously presented models can be used as features. **Retrieval model based features** (Φ_R) are the scores computed in a fashion similar to the *Nearest Context*

model. Here we define three features,

$$\begin{aligned}\Phi_R(\langle u_i, context_j \rangle) &= \{retrieval_score, context_sim@best_utt_match, \\ &\quad utt_sim@best_context_match\} \\ retrieval_score &= \max_{k=1}^{|L|} Sim(context_j, context_k) \cdot Sim(u_i, u_k) \\ context_sim@best_utt_match &= Sim(context_j, context_b) \text{ where, } b = \arg\max_{k=1}^{|L|} Sim(u_i, u_k) \\ utt_sim@best_context_match &= Sim(u_i, u_b) \text{ where, } b = \arg\max_{k=1}^{|L|} Sim(context_j, context_k)\end{aligned}$$

$Sim(u_x, u_y)$ is a cosine similarity function for tf-idf weighted vector space representations of utterances and $Sim(context_a, context_b)$ is the same function from *Nearest Context* model.

Topic based feature (Φ_T) tracks the topic similarity between the topic of the dialogue context and the response utterance. Topics are tracked in the same way as in *Segmented Nearest Context* model. Each information state (IS) consists of a topic signature which can be viewed as a boolean vector representing mentions of topics.

$$\begin{aligned}\Phi_T(\langle u_i, context_j \rangle) &= \{topic_similarity\} \\ topic_similarity &= cosine(IS_i, IS_j)\end{aligned}$$

where, IS_i is the topic and is part of $context_i$ which is the context associated with the utterance u_i .

Figure 4.15 shows an example context and utterance and the corresponding feature vector.

$context_j$	$\begin{bmatrix} u_{j(-2)} \\ u_{j(-1)} \end{bmatrix}$	\vdots	
		Doctor	you are the threat i need protection from you
		Captain	no no
			you do you do not need protection from me
			i am here to help you
			uh what i would like to do is move your your clinic to a safer location
			and uh give you money and medicine to help build it
<hr/>			
$utterance$	$[u_i]$	Doctor	i have no way of moving
<hr/>			
$\Phi_S(\langle u_i, context_j \rangle) = \{$			
$cross_term(-2, "moving", "need") = 1,$			
$common_term(-2, "i") = 1,$			
$common_term_count(-2) = 1, unique_common_term_count(-2) = 1,$			
$cross_term(-1, "moving", "give") = 1,$			
$common_term(-1, "i") = 1, common_term(-1, "no") = 1,$			
$common_term_count(-1) = 2, unique_common_term_count(-1) = 2,$			
$retrieval_score = 0.198, context_sim@best_utt_match = 0.198,$			
$utt_sim@best_context_match = 0,$			
$topic_similarity = 0.667 \}$			

Figure 4.15: Features extracted from a context ($context_j$) and a response utterance (u_i)

The perceptron model presented here allows novel combinations of resources such as combining surface text transcripts with information state annotations for tracking topics in the conversation. As compared to the generative cross-lingual relevance model approach, the perceptron model is a discriminative model. It is also a parametric model and the inference requires linear time with respect to the size of candidate utterances ($|GEN(context)|$) and the number of features ($|\bar{\alpha}|$). Although, computing some of the features themselves (e.g., Φ_R features) requires linear time with respect to the size of the training data. The perceptron model can rank an arbitrary set of utterances given a dialogue context. But some

of the features (e.g., *topic-similarity*) require that the utterance $u_i (u_i \in |\text{GEN}(\text{context})|)$ be associated with a known context (context_i). For all our models we use $\text{GEN}(\text{context}) = U_{\text{train}}$.

We have implemented three different variants of the perceptron model based on the choice of features used. *Perceptron(surface)* model uses only surface text features ($\Phi = \Phi_S$). The other two models are *Perceptron(surface+retrieval)* where $\Phi = \Phi_S \cup \Phi_R$ and *Perceptron(surface+retrieval+topic)* where $\Phi = \Phi_S \cup \Phi_R \cup \Phi_T$.

	Original Human-Human Dialogue	Model response
Captain	hello how are you doing sir	
Doctor	what do you want i have patients waiting for me	hello what can i do for you mr
Captain	sir i come by to offer you some assistance ah on behalf of my commander ah we want to try to support you in this hospital and ah improving the health and welfare of the civilians that you have under your care	
Doctor	well this is your opinion	i see captain is this a is this a suggestion from your commander
Captain	i am sorry sir could you say again	
Doctor	well this is your opinion	you are the threat i need protection from you

Figure 4.16: Example interaction for *Perceptron* model in static context setting. The second column shows the original human-human dialogue and the third column shows the *Perceptron(surface)* model’s response for the corresponding system turn.

4.7 Evaluation of Dialogue Models

In Chapter 1 Section 1.4, we presented the decisions that are involved in evaluating dialogue models for virtual humans. Here we present an evaluation of the dialogue models described in Section 4.6. Since the goal for virtual humans is to be as human-like as possible, a suitable evaluation metric for virtual human dialogue systems is how appropriate or human-like the responses are for a given dialogue context. The evaluation reported in this chapter employs human judges. We set up a simple subjective 5-point likert scale for rating appropriateness – 1 being a very inappropriate non-sensical response and 5 being a perfectly appropriate response. Due to the cost associated with human evaluations, we did not evaluate every model in every possible evaluation setting. We evaluated four dialogue models: *Random*, *Nearest Context*, *Segmented Nearest Context* and *Segmented Random* in *dynamic context* setting and five dialogue models: *Nearest Context*, *Cross-lingual Relevance Model*, *Perceptron(surface)*, *Perceptron(surface+retrieval)* and *Perceptron(surface+retrieval+topic)* in *static context* setting. We also evaluated one *human* upper baseline model in *dynamic context* setting and two human-level upper baseline models: *Wizard Random* and *Wizard Max Voted* in *static context* setting.

4.7.1 Dynamic Context

In *dynamic context* evaluation, the dialogue model is used for generating the response utterances as well as the dialogue contexts with respect to which the subsequent responses are evaluated. The resulting dialogue is generated through an interactive process between a human user and the dialogue model. We built dialogue models for playing the role of the *doctor* in SASO-ST domain using all the training data. These dialogue models were used to build virtual human dialogue systems. The dialogue systems are designed to avoid repetitions in dialogue and provide second-best response if the top-ranked response from the dialogue model was same as the immediately preceding system response.

4.7.1.1 Dynamic Context Evaluation by Dialogue Participants

We evaluated four dialogue models: *Random*, *Nearest Context*, *Segmented Nearest Context* and *Segmented Random* in dynamic context setting where the judges were also dialogue participants (Gandhe and Traum, 2007b). Input and output modality was limited to typed text only and the turns were strictly alternated. We asked volunteers to conduct conversations with the simulated doctor. These volunteers had two roles - as a participant in the negotiation conversation and also as a judge of the responses from the doctor. The interface shown in Figure 4.17 allows the volunteers to judge the doctor's response on a scale of 1 to 5 for appropriateness. We had six volunteers, each interacting once with each of the four dialogue models. The presentation order of the models was balanced.

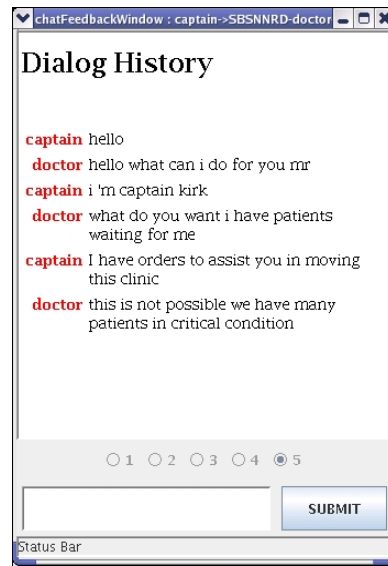


Figure 4.17: A screen-shot of the user interface for dynamic context evaluation by the dialogue participants.

The average ratings for each type of model are summarized in Table 4.2. We performed Wilcoxon rank sum test for comparing these models. *Segmented Nearest Context* was the best performing model. It was significantly better than *Nearest Context* ($p < 0.05$) and *Segmented Random* ($p < 0.05$) and *Random* ($p < 0.001$). *Segmented Random* was significantly better than *Random* ($p < 0.05$). All other differences are not statistically significant at the 5 percent level. Figure 4.18 shows a bar plot for the average appropriateness of the four models. The error bars indicate 95% confidence intervals.

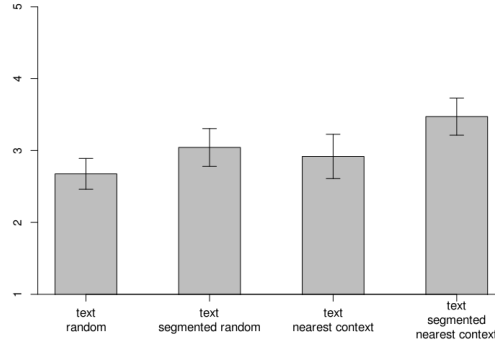


Figure 4.18: Average appropriateness levels for the four dialogue models evaluated by the dialogue participants themselves in dynamic context setting.

	Without Segments	With Segments
	<i>Random</i>	<i>Segmented Random</i>
Without	avg 2.676	avg 3.043
local	stddev 1.276	stddev 1.293
context	size 136	size 93
	<i>Nearest Context</i>	<i>Segmented Nearest Context</i>
With	avg 2.918	avg 3.472
local	stddev 1.557	stddev 1.370
context	size 98	size 108

Table 4.2: Results for the four dialogue models as evaluated by the dialogue participants themselves in dynamic context setting.

4.7.1.2 Dynamic Context Evaluation by Bystanders

We evaluated the same four models: *Random*, *Nearest Context*, *Segmented Nearest Context* and *Segmented Random* in dynamic context setting and with the help of bystanders (Gandhe and Traum, 2007a). There were two settings based on what input and output modality was used during the interactions with the virtual human character. The evaluators were presented with a web-interface where they could provide appropriateness ratings for responses given the dialogue contexts. The evaluators only saw dialogue transcripts.

Typed Text Modality

Two judges evaluated the dialogues collected from the previous experiment as described in section 4.7.1.1. From that experiment, we had 24 dialogues where both the input and output modalities are typed text. We had the evaluators judge the doctor’s utterances for appropriateness on the same scale of 1 to 5. The results are summarized in Table 4.3.

We performed Wilcoxon rank sum test to check whether the differences are statistically significant. Every other system was significantly better than *Text Random* ($p < 0.001$). *Text Nearest Context* was significantly better than *Text Segmented Random* ($p < 0.05$). *Text Segmented Nearest Context* was significantly better than *Text Segmented Random* ($p < 0.01$) but not significantly better than *Text Nearest Context*.

Model	#Utts	Avg. appropriateness		Appropriateness (All judges)	
		Judge1	Judge2	Avg	SEM
<i>Text Random</i>	141	2.75	1.86	2.30	0.070
<i>Text Segmented Random</i>	96	3.52	2.47	2.99	0.088
<i>Text Nearest Context</i>	103	3.85	2.60	3.23	0.092
<i>Text Segmented Nearest Context</i>	113	3.96	2.78	3.37	0.091

Table 4.3: Results for the four dialogue models as evaluated by judges not participating in the conversation (bystanders) in dynamic context setting.

Speech Modality

To understand the effect of speech as input modality, we collected dialogues from spoken interactions with an embodied character. The animated body, speech recognition, speech synthesis and gesture production components from the SASO-ST system (Traum et al., 2005) were used. We asked four volunteers to conduct conversations with the doctor character in two settings where the dialogue model used was either *Random* (labeled *Speech Random*) or *Segmented Nearest Context* (labeled *Speech Segmented Nearest Context*). Each volunteer talked to both versions and presentation order was balanced. These spoken dialogues were later transcribed and the word error rate for the *Speech Random* dialogues was 0.52, while for *Speech Segmented Nearest Context* it was 0.41. Evaluators saw the transcriptions and not the recognized speech. The responses from the models were based on the recognized speech.

Speech Segmented Nearest Context performed significantly better than *Speech Random* ($p < 0.001$). We performed a one-sided Wilcoxon rank sum test to see if speech modality degrades the performance. *Text Segmented Nearest Context* performs better than *Speech Segmented Nearest Context* ($p < 0.05$). All other differences are not statistically significant at the 5 percent level.

Apart from evaluating these four dialogue models, we also evaluated a human-level upper baseline. The human-human dialogues from our training corpus could be thought of as dialogues that were collected in dynamic context setting using a *human* dialogue model. We chose 4 such human-human dialogues. We had the same two evaluators judge the doctor’s utterances for appropriateness on the same scale of 1 to 5. Evaluators used only the text transcriptions to make their judgments. Table 4.4 summarizes the results.

Model	#Utts	Avg. appropriateness		Appropriateness (All judges)	
		Judge1	Judge2	Avg	SEM
<i>Speech Random</i>	88	3.14	2.06	2.60	0.102
<i>Speech Segmented Nearest Context</i>	99	3.47	2.78	3.13	0.096
<i>Human-Human</i>	91	4.38	3.36	3.87	0.074

Table 4.4: Results for two dialogue models in embodied character settings and speech as input modality along with the evaluation of human-human dialogues performed by bystanders in dynamic context setting.

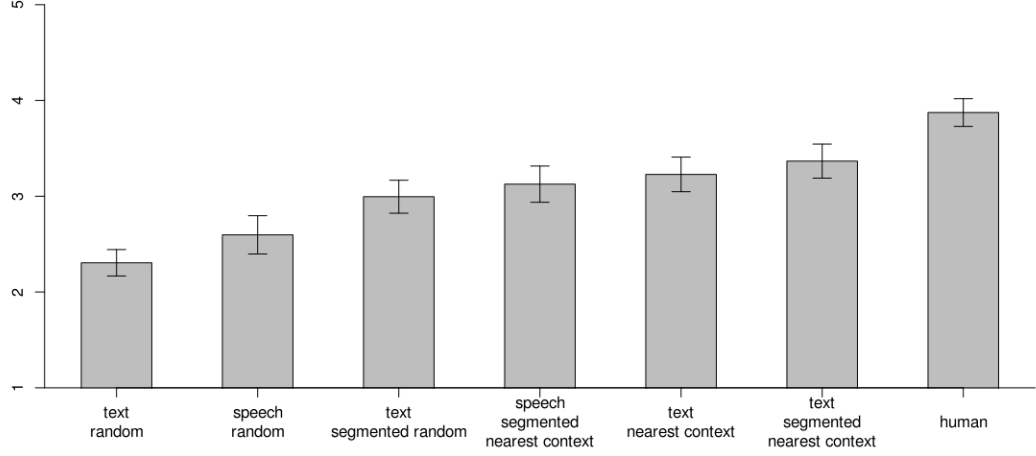


Figure 4.19: Average appropriateness levels for various dialogue models.

Figure 4.19 presents the average appropriateness levels for all the different models in different settings including speech and text modalities. The error bars show 95% confidence intervals based on standard error around the mean. *Human-Human* dialogues were significantly better than all the models ($p < 0.001$). The best performing model *Text Segmented Nearest Context* can achieve up to 68% of human-level performance above the *Text Random* baseline.

$$\frac{\text{Text Segmented Nearest Context} - \text{Text Random}}{\text{Human-Human} - \text{Text Random}} = \frac{3.37 - 2.3}{3.87 - 2.3} = 0.68$$

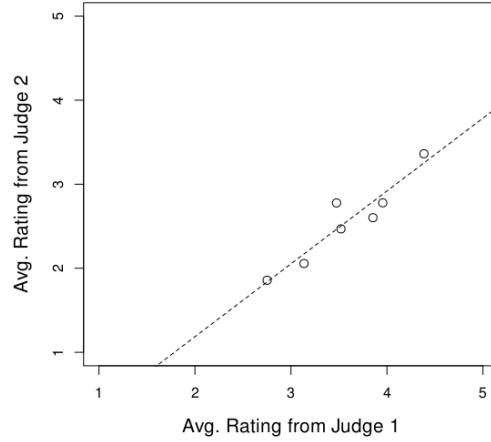


Figure 4.20: Scatter plot of average appropriateness levels for different models as judged by two judges.

Figure 4.20 shows the scatter plot of average ratings for all the models as judged by our two judges. The linearity of the plot suggests high inter-rater agreement, even though one rater tended to give much higher scores across the board. As a measure of inter-rater agreement we calculated the Pearson's correlation coefficient for average appropriateness levels. It is quite high at 0.94 ($p < 0.01$).

4.7.2 Static Context

In *Static Context* evaluation the dialogue contexts are fixed and are not affected by the dialogue model being evaluated. Static context evaluation is particularly well suited for comparative evaluation of dialogue models. All the dialogue models being evaluated receive the same set of contexts as input. In contrast, for dynamic context setting the set of contexts used as input would be different for different models. Thus the static context setting allows for dialogue models to be compared more easily than in dynamic context setting as less data needs to be collected. These dialogue contexts are extracted from actual in-domain human-human dialogues. For every turn whose role is to be played by the system, we predict the most appropriate response in place of that turn given the dialogue context. This response chosen by the dialogue model is then rated for appropriateness using the same scale of 1 to 5 by an evaluator. In static context evaluation, the evaluator is always a bystander as he doesn't take part in creating the original human-human dialogue.

We built five dialogue models to play the role of the doctor in SASO-ST domain, viz.: *Nearest Context* (section 4.6.2), *Cross-lingual Relevance Model* (section 4.6.5) and three *perceptron* models (section 4.6.6) with different feature sets. These dialogue models are evaluated using 5 in-domain human-human dialogues from the training data (2 dialogues from roleplays and 3 from WoZ corpus, referred to as test dialogues). A dialogue model is trained in a leave-one-out fashion where the training data consists of all dialogues except the one test dialogue that is being evaluated. A dialogue model trained in this fashion is then used to predict the most appropriate response for every context that appears in the test dialogue. This process is repeated for each test dialogue and for each dialogue model being evaluated. In this evaluation setting, the actual response utterance found in the original human-human dialogue may not belong to the set of utterances being ranked by the dialogue model. We also compare these five dialogue models with two human-level upper baselines.

4.7.2.1 Wizard Data Collection

In order to establish an upper baseline for human-level performance in the static context evaluation task, we conducted a wizard data collection. We asked human volunteers (wizards) to perform a similar task to that performed by the dialogue models being evaluated. The wizard is presented with a set of utterances (U_{train}) and is asked to select a subset from these that will be appropriate as a response for the presented dialogue context. Compared to this, the task of the dialogue model is to select a single most appropriate response for the given context.

We have a reason to believe that for a given dialogue context there are several appropriate response utterances at the surface text level. DeVault et al. (2011b) carried out a similar wizard data collection but at the dialogue act level, where wizards were asked to select only one response dialogue act for a given dialogue context. Their findings suggest that there are several valid response dialogue acts for a given dialogue context. A specific dialogue act can be realized in several ways at the surface text level. In our setting the dialogue models work at the surface text level and hence the wizards were asked to select a subset of surface text utterances which would be appropriate responses. Each wizard was requested to select somewhere between 5 to 10 (at-least one) appropriate responses for each dialogue context. Four wizards participated in this data collection with each wizard selecting responses for the contexts from the same five human-human test dialogues. The set of utterances to choose from (U_{train}) for every test dialogue was built in the same leave-one-out fashion as used for evaluating the implemented dialogue models.

Apart from establishing an upper baseline of human-level performance, this wizard data can also be used for developing an evaluation understudy for static context evaluation (see Chapter 5 Section 5.1).

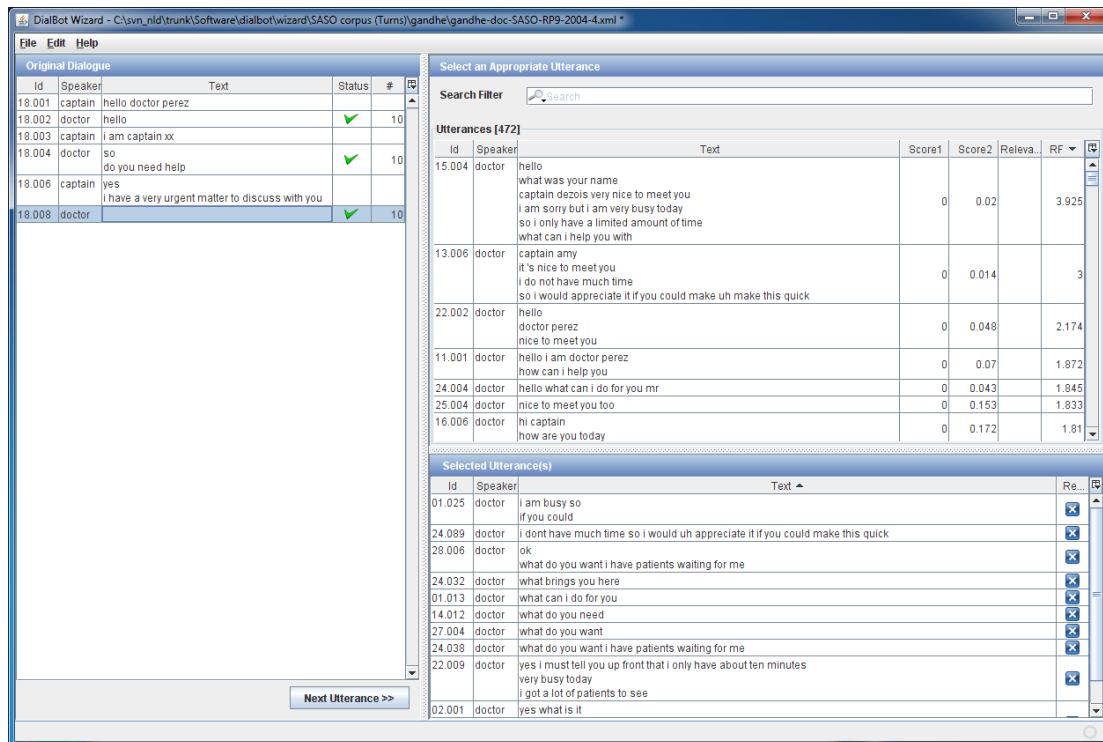


Figure 4.21: A screenshot of the interface for the wizard data collection.

With a goal of rapid and cost-efficient evaluation of virtual human dialogue systems, we need to employ *non-experts* to collect this wizard data at reduced cost. In order to help wizards select the set of appropriate response utterances from U_{train} ($|U_{train}| \approx 500$) we built a wizard data collection tool.

Figure 4.21 shows the screenshot of the interface presented to the wizards. The left panel of the interface shows the dialogue context from the original human-human dialogue. The top right panel lists all possible responses and the bottom right panel shows the utterances selected by a wizard as appropriate responses for the given dialogue context. As an additional help for wizards to explore the set of all possible utterances U_{train} we provide the ability to rank the utterances by various scores. *Score1* is the score calculated using the *Nearest Context* model. *Score2* is surface text similarity computed as METEOR score between the candidate utterance and the actual response utterance present at that location in original human-human dialogue. Wizards were not aware of what score1 and score2 indicate. They were just told that they may find these scores useful. Wizards can search the set of utterances for specific keywords and *Relevance* column shows this relevance score for the search string entered by the wizards. The last column *RF* stands for relevance feedback and ranks the utterances based on what has already been chosen by the wizard. This allows wizards to easily find paraphrases of already selected response utterances. Wizards were instructed in how to use the search and relevance feedback functionalities.

There are a total of 89 dialogue contexts where the next turn belongs to *doctor*. Figure 4.22 shows the histogram for the number of utterances selected as appropriate responses by four wizards. As expected, wizards frequently chose multiple utterances as appropriate responses (mean = 7.80, min = 1, max = 25).

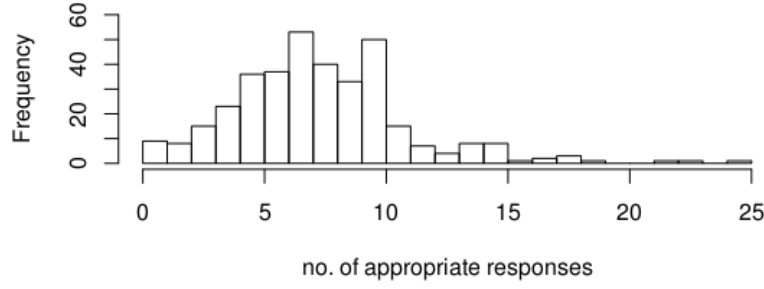


Figure 4.22: A Histogram for the number of selected appropriate response utterances.

To get an idea about how much the wizards agree among themselves for this task, we calculated the overlap between the utterances selected by a specific wizard and the utterances selected by another wizard or a set of wizards. Let U_c^T be a set of utterances selected by a wizard T for a dialogue context c . Let R be a set of wizards ($T \notin R$) and U_c^R be the union of sets of utterances selected by the set of wizards (R) for the same context c . Then we define the following overlap measures,

$$\begin{aligned} \text{Precision}_c &= \frac{|U_c^T \cap U_c^R|}{|U_c^T|} & \text{Recall}_c &= \frac{|U_c^T \cap U_c^R|}{|U_c^R|} \\ \text{Jaccard}_c &= \frac{|U_c^T \cap U_c^R|}{|U_c^T \cup U_c^R|} & \text{Dice}_c &= \frac{2|U_c^T \cap U_c^R|}{|U_c^T| + |U_c^R|} \end{aligned}$$

$$\text{Meteor}_c = \frac{1}{|U_c^T|} \sum_{u_t \in U_c^T} \text{METEOR}(u_t, U_c^R) \quad \forall u_t \in U_c^T$$

We compute the average values of these overlap measures for all contexts and for all possible settings of test wizards and reference wizards. Table 4.5 shows the results with different values for the number of wizards used as reference.

	Precision	Recall	Jaccard	Dice	Meteor
One Reference	0.145	0.145	0.077	0.141	0.290
Two References	0.244	0.134	0.093	0.170	0.412
Three References	0.311	0.121	0.094	0.171	0.478

Table 4.5: Overlap measures indicating inter-wizard agreement

Precision can be interpreted as the probability that a response utterance selected by a wizard is also considered appropriate by at least one other wizard. Precision rapidly increases along with the number of reference wizards used. This happens because the size of the set U_c^R steadily increases with more reference wizards. Figure 4.23 shows this observed increase and the expected increase if there were no overlap between the wizards. The near-linear increase in $|U_c^R|$ suggests that selecting appropriate responses is a hard task and may require a lot more than four wizards to achieve convergence.

This data collected from wizards is used to build two human-level upper-baseline models for the task of selecting a response utterance given a dialogue context:

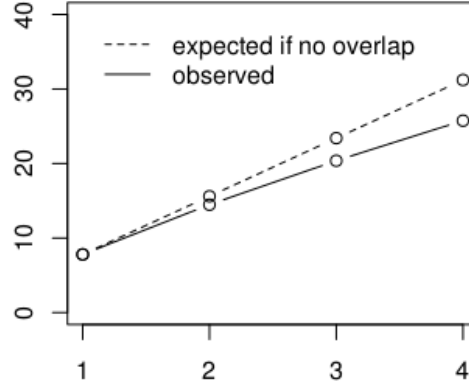


Figure 4.23: Avg. cardinality of the set U_c^R – union of sets of utterances selected as appropriate responses by wizards R for different values of $|R|$

Wizard Max Voted model returns the response which gets the maximum number of votes from the four wizards. Ties are broken randomly.

Wizard Random model returns a random utterance from the list of all utterances marked as appropriate by any of the wizards.

4.7.2.2 Comparative Evaluation of Models

We performed a static context evaluation using four judges for the two human-level baselines (*Wizard Random* and *Wizard Max Voted*) and the above-mentioned five dialogue models (*Nearest Context*, *Cross-lingual Relevance Model* and three *perceptron* models). The *Perceptron(surface)* model uses features based on surface text only. *Perceptron(surface+retrieval)* model uses features based on surface text as well as retrieval score features. *Perceptron(surface+retrieval+topic)* model uses additional features based on topic. We tune the parameters used for the perceptron models based on the automatic evaluation method, R_{weak} , which will be described in section 5.1.1. As a result, *Perceptron(surface)* model was trained using 30 iterations and *Perceptron(surface+retrieval)* using 20 iterations and *Perceptron(surface+retrieval+topic)* was trained using 25 iterations. For all perceptron models we used $threshold_x = threshold_y = threshold_{xy} = 3$.

For a comparative evaluation of dialogue models, we need an evaluation setup where judges could see the complete dialogue context along with the response utterances generated by the dialogue models to be evaluated. In this setup, we show all the response utterances next to each other for easy comparison and we do not show the actual response utterance that was encountered in the original human-human dialogue. We built a web interface for collecting appropriateness ratings that addresses the above requirements. Figure 4.24 shows the web interface used by the four judges to evaluate the appropriateness of response utterances for given dialogue context. The appropriateness was rated on the same scale of 1 to 5. The original human-human dialogue (roleplay or WoZ) is shown on the left hand side and the response utterances from different dialogue models are shown on the right hand side. In cases where different dialogue models produce the same surface text response only one candidate surface text is shown to judge. Once the judge has rated all the candidate responses they can proceed to the next dialogue context. This setting allows for comparative evaluation of different dialogue models. The presentation order of responses from different dialogue models is randomized. Two of the judges also performed the role

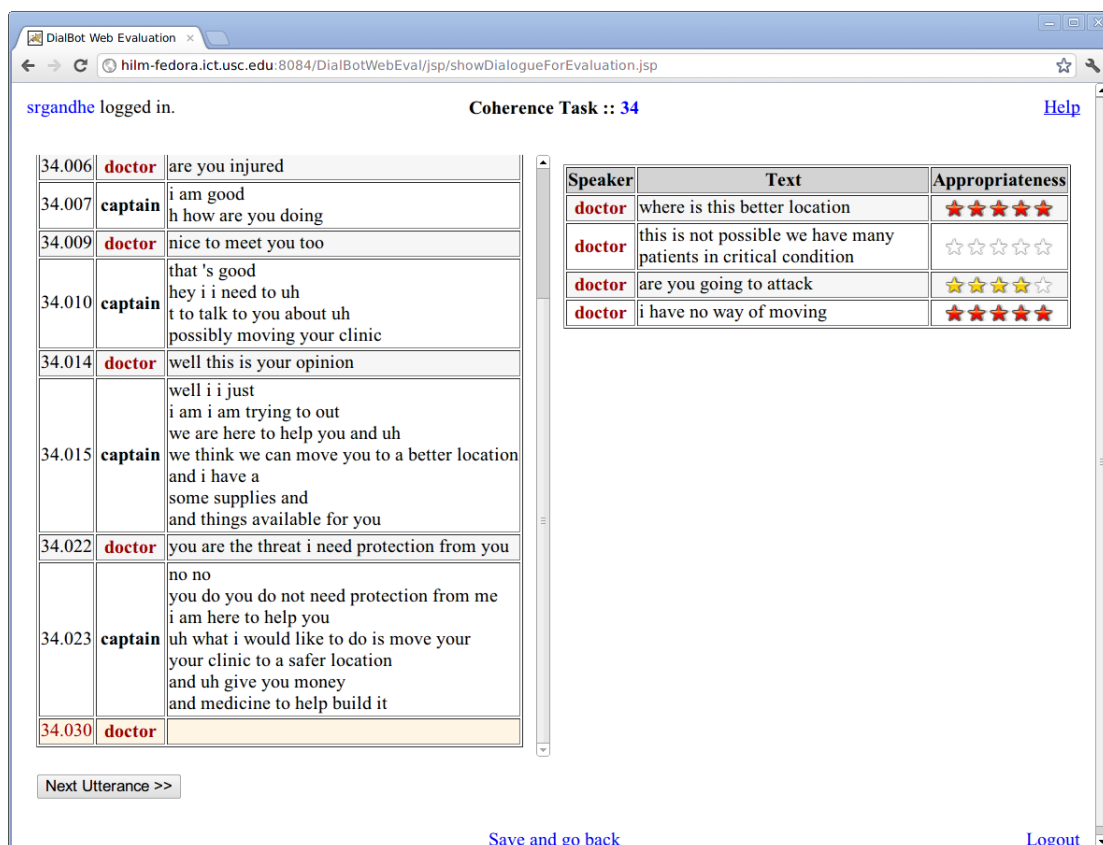


Figure 4.24: Screenshot of the user interface for static context comparative evaluation of dialogue models

of the wizards in our wizard data collection as outlined in section 4.7.2.1, but the wizard data collection and the evaluation tasks were separated by a period of over 3 months.

Table 4.6 shows the results of our comparative evaluation for each judge and averaged over all judges. We also computed inter-rater agreement for individual ratings for all response utterances using Krippendorff's α (Krippendorff, 2004). There were a total of $n = 397$ distinct response utterances that were judged by the evaluators. The Krippendorff's α for all four judges was 0.425 and it ranges from 0.359 to 0.495 for different subsets of judges. The value of α indicates that the inter-rater agreement is substantially above chance ($\alpha > 0$). The lower range of α values indicates that judging appropriateness is a hard task even for human judges. Although there is low inter-rater agreement at the individual response utterance level there is high agreement at the dialogue model level. Pearson's correlation between the average appropriateness for different dialogue models ranges from 0.928 to 0.995 for different pairs of judges.

Figure 4.25 shows a barchart of the appropriateness ratings for seven dialogue models averaged over all four judges and all contexts. The error bars show 95% confidence intervals. We performed a paired Wilcoxon test to check for statistically significant differences in different dialogue models. *Wizard Max Voted* is significantly more appropriate than all other models ($p < 0.001$). *Wizard Random* is significantly more appropriate than *Cross-lingual Relevance Model* ($p < 0.05$) and significantly more appropriate than the three perceptron models as well as *Nearest Context* model ($p < 0.001$). *Cross-lingual Relevance*

Model	#Utts	Avg. appropriateness				Appropriateness (All judges)	
		Judge 1	Judge 2	Judge 3	Judge 4	Avg	stddev
<i>Nearest Context</i>	89	4.12	3.98	3.40	3.53	3.76	1.491
<i>Perceptron(surface)</i>	89	3.97	4.11	3.51	3.62	3.80	1.445
<i>Perceptron (surface+retrieval)</i>	89	4.26	4.12	3.51	3.72	3.90	1.414
<i>Perceptron (surface+retrieval+topic)</i>	89	4.21	4.09	3.51	3.57	3.85	1.433
<i>Cross-lingual Relevance Model</i>	89	4.28	4.31	3.70	3.91	4.05	1.314
<i>Wizard Random</i>	89	4.55	4.55	4.03	4.16	4.32	1.153
<i>Wizard Max Voted</i>	89	4.76	4.84	4.40	4.52	4.63	0.806

Table 4.6: Offline comparative evaluation of dialogue models.

Model is significantly more appropriate than *Nearest Context* ($p < 0.01$). All other differences are not statistically significant at the 5 percent level. Table 4.7 shows pairwise comparisons among the seven dialogue models in static context setting. It includes details about how often a given pair of dialogue models produce the same response for a given context.

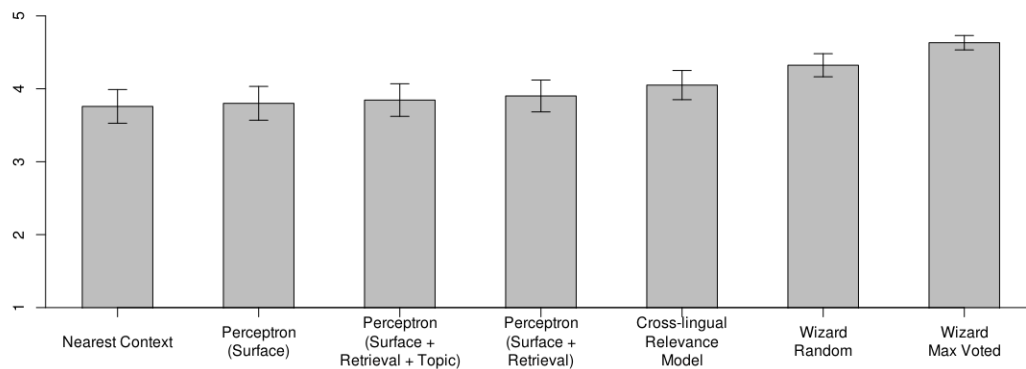


Figure 4.25: Results of static context comparative evaluation of dialogue models.

4.8 Discussion

The *topic* information being used for the *Segmented Nearest Context* and *Perceptron(surface+retrieval+topic)* models is currently based on simple keyword spotting. This method has the advantage of automatically tagging the topic signature for dialogue contexts. It is simple enough for *non-experts*, all that is required is to list the key concepts along with the words used to signal them. But it is possible to misrecognize the topic signature for a dialogue context. This may happen due to the loose coupling between key concepts and the representative words. E.g., if the captain says the phrase “shifting the clinic”, it will not register the concept of *move* as ‘shift’ is not one of the words listed under the concept *move* (See Figure 4.11).

	<i>Nearest Context</i>	<i>Perceptron(surface)</i>	<i>Perceptron (surface+retrieval)</i>	<i>Perceptron (surface+retrieval +topic)</i>	<i>Cross-lingual Relevance Model</i>	<i>Wizard Random</i>	<i>Wizard Max Voted</i>
<i>Nearest Context</i>							
<i>Perceptron(surface)</i>	41 $\frac{02}{12}$ 34						
<i>Perceptron (surface+retrieval)</i>	07 $\frac{75}{02}$ 05	36 $\frac{02}{12}$ 39					
<i>Perceptron (surface+retrieval +topic)</i>	07 $\frac{74}{00}$ 08	35 $\frac{02}{10}$ 42	01 $\frac{81}{03}$ 04				
<i>Cross-lingual Relevance Model</i>	33 $\frac{33}{06}$ 17 **	38 $\frac{03}{11}$ 37	32 $\frac{28}{06}$ 23	33 $\frac{29}{05}$ 22			
<i>Wizard Random</i>	51 $\frac{04}{14}$ 20 ***	51 $\frac{02}{12}$ 24 ***	50 $\frac{04}{14}$ 21 ***	52 $\frac{04}{13}$ 20 ***	41 $\frac{04}{15}$ 29 *		
<i>Wizard Max Voted</i>	55 $\frac{07}{16}$ 11 ***	55 $\frac{06}{15}$ 13 ***	51 $\frac{08}{18}$ 12 ***	52 $\frac{07}{18}$ 12 ***	50 $\frac{04}{18}$ 17 ***	39 $\frac{10}{27}$ 13 ***	

$a \frac{c}{d} b$ a : number of times the *row* model is judged as more appropriate than *column*.
 b : number of times the *column* model is judged as more appropriate than *row*.
 c : number of times both the models produce responses with exactly same surface text for a
 given context. d : number of times the responses from both models are judged as equally appropriate
 excluding the cases denoted by c. *** : $p < 0.001$ ** : $p < 0.01$ * : $p < 0.05$

Table 4.7: Pairwise comparisons for evaluating appropriateness in static context setting along with statistical significance.

Similarly the phrase “moving the chair” can erroneously trigger the *move* concept. Simple keyword spotting is not adequate for accurately detecting dialogue topic segment transitions. In spite of this, for the SASO-ST domain the simple keyword spotting approach for annotating topic signature works well.

During the dynamic context evaluations performed by the dialogue participants, the subjective feedback we received from the users was that the *Segmented Nearest Context* model performs well enough to conduct meaningful dialogues. The reason behind this success stems from the fact that these conversations are restricted as they have to follow the story line. Also since the task structure is shallow, just identifying the correct dialogue segment helps quite a bit. Also using the *selection* approach rather than *generation* adds more richness and naturalness to the replies from the virtual human, making it more believable.

The way these unsupervised dialogue models differ from information-state based dialogue systems is that there is no need for rule authoring or corpus annotation. Theoretically information state based systems can achieve arbitrary levels of perfection given enough rules are authored for it. But rule authoring and corpus annotation still remains a practical limiting factor for such systems. We do miss the deep understanding that an information-based system gives, such as principled connection to emotions and plan reasoning. Although this can be mitigated by allowing light annotations of the corpus similar to the topic signature annotations done for avoiding unwarranted presuppositions.

When compared to general purpose chat-bot systems like Eliza or Alice, our domain of interaction is well defined. E.g., Chat-bot systems have to be ready to talk about a variety of topics, including favorite movies, sports etc., whereas our system just has to know about the negotiation scenario between the captain and the doctor.

The granularity of the utterances is another issue. Figure 4.26 shows an example where the utterance refers to security issues by mentioning the problems of the blocked roads but also talks about other things not mentioned in the preceding context. This makes the utterance less coherent. We need to split this turn into appropriate units and select only that which will be a coherent response. Another issue is the amount of speech disfluencies and overlaps involved in roleplay dialogues. This can be mitigated by pre-processing the corpus to remove such artifacts (Jonsson and Dahlback, 2000) or by collecting the roleplays in a more restricted interface such as text chat.

4.9 Applications

Our implemented dialogue models can be built from un-annotated corpora of in-domain human-human dialogues. They are appropriate for applications where one needs rapid prototyping of dialogue systems designed with a human metaphor in mind (like virtual humans). The only application specific resource required for the models is the dialogue corpus. It is always worth collecting this dialogue corpus to refine the domain constraints and to create acoustic and language models for speech recognition purposes. Our dialogue models operate on the surface text level and do not involve rule authoring. They are robust against speech recognition errors and avoid the brittleness of rule-based systems. Here we list some of the ways these models can be used.

- Surface text based methods for dialogue modeling have been shown to be useful for reactive behavior (Leuski et al., 2006; Artstein et al., 2008). When presented with user input, virtual humans can either deliberate or react or both. Deliberation requires reasoning about how the user’s utterance fits into the task being carried out, the beliefs and desires of the virtual human, how best to achieve its goals, etc. Reactions on the other hand can be more directly based on the surface form of the utterance. This allows the dialogue models to come up with an appropriate response at the reactive level.

	Speaker	Utterance	Rating
D1	Doctor	uh i'm sorry	5
		what was your name	
C2	Captain	i'm captain kirk	4
D3	Doctor	captain it's nice to meet you	
		i don't have much time so i would appreciate it if you could make uh make this quick	
C4	Captain	ok doctor.	2
		I want to talk about the security of the area this hospital is located in.	
D5	Doctor	well they i don't know that there is very good uh anywhere	
		this this one seems to be full of patients who i need to be treating	
		but the problem is the roads are not always clear	
		and it's necessary uh you know that we are able to get supplies and and	
		we're not	
C6	Captain	yes doctor,	
		it is hard to get supplies here, because of the danger.	
		We do have access to supplies, but it will be hard to bring them here.	

Figure 4.26: Illustration of the problem due to the granularity of the utterance. This dialogue is generated by using *Segmented Nearest Context* model and shows the evaluation by the dialogue participant in dynamic context setting. Last utterance from the doctor gets a low rating.

- Since our models are symmetrical with respect to the speaker (whether it is user or system), These models can be used to come up with user's utterances as well. This can be used as a user simulation module. User simulations can be used for testing the dialogue systems (Eckert et al., 1997) or even to improve them through reinforcement learning (Levin et al., 1997). The predictions on the user's utterances can also be used to improve the speech recognition by adaptive language modeling.
- The proposed dialogue models can be used in conjunction with traditional information-state based dialogue models. The traditional models can come up with a ranked list of the responses given the dialogue context. Our dialogue models can be used to rescore such alternatives and select the most appropriate responses. There might also be situations where traditional dialogue system cannot come up with a response at all, in which case our models can help select one of the responses from already seen corpus.
- In simulation training scenarios like (Traum et al., 2008c), these models can be used for cursory virtual characters (Jan and Traum, 2005) which need to be built rapidly and with minimal amount of effort. Our dialogue models do not have rich-knowledge structures for the information state, virtual humans built by using these dialogue models are not *explainable*. The dialogue mechanism itself cannot be used to explain the dialogue behavior. But for the cursory virtual characters such explainability may not be required.
- These dialogue models can be used to improve the collection of a corpus through WoZ studies as well. The data collected through roleplay activities can be used to model WoZ dialog systems. The ranked list of system utterances can be presented to a wizard for selection. This should make the process of choosing the system utterances more efficient for the wizard. Wizard's selections can be used as evaluation. Moreover since no additional annotations are needed the dialogue systems can be improved instantly by adding the collected WoZ data back into the dialogue models.

- These dialogue models have an application in Augmentative and Alternative Communication (AAC) technology. The goal for AAC is to help individuals with speech and language impairments communicate. One of the ways to achieve the goal is to build predictive models for language production. Recently, there have been efforts towards whole-sentence prediction in conversational settings (Mitchell and Sproat, 2012). We believe our models once trained with individual specific corpus about personal information such as family, hobbies, likes/dislikes, etc. will prove useful in AAC.

4.10 Conclusion

One of the contributions of this thesis is to develop flexible architectures that allow novel combinations of different types of resources, such as surface text transcripts and information state annotations. In this chapter, we have implemented unsupervised dialogue models that work primarily at the surface text level. Our dialogue models employ the *selection* approach for formulating a response. We have also presented a method to verify the viability of the selection approach for given domain and available corpus. The virtual human dialogue system based on such models can be bootstrapped from only the resource of unannotated in-domain human-human corpus. Resources such as information state annotations for topics, etc. can be incrementally added to the dialogue model as they become available. We have built dialogue models based on simple information retrieval techniques (*Nearest Context*) as well as advanced text-to-text methods (*Perceptron* and *Cross-lingual Relevance Model*).

We have evaluated these models in different settings – *dynamic context* or *static context*, and with different evaluator choices – dialogue participants themselves or bystanders, and with different modalities – typed text or speech. In dynamic context evaluation by dialogue participant themselves, the *Text Segmented Nearest Context* model which uses the information state annotations with topic signatures in addition to the surface text of the context performs significantly better than the model *Text Nearest Context* which uses context information alone. In dynamic context evaluation by bystanders, the best performing model *Text Segmented Nearest Context* can achieve upto 68% of human-level performance above the *Text Random* baseline. Speech recognition errors significantly degrade the performance of *Speech Segmented Nearest Context* model as compared to the *Text Segmented Nearest Context* model. In static context evaluation, the advanced text-to-text models which use the utterance content information in addition to context information perform better than the simpler *Nearest Context* model which uses context information alone. The *Cross-lingual Relevance Model* model performs significantly better than the *Nearest Context* model. We have also shown that the *Perceptron* model easily allows novel combinations of resources by simply changing the set of features used. This allows us to better understand the relative utility of different resources and the resulting information can be used to optimize the cost and/or performance of the dialogue system. Resulting dialogue models can be used in a virtual human dialogue system along with several other applications as listed in section 4.9.

Chapter 5

Automatic Evaluation for Dialogue Models

One of the goals of this thesis is rapid and cost-efficient evaluation of dialogue models, such as those described in Chapter 4. These models employ the *selection* approach for response formulation and can be trained with novel combinations of dialogue system resources.

For interactive applications of natural language technology, such as dialogue models, the best way to evaluate a novel algorithm or a model would be to employ it in a real system and allow users to interact with it. As per the terminology introduced in Chapter 1 Section 1.4, the ideal setting would be *dynamic context* evaluation. As described in Chapter 4, for evaluating unsupervised dialogue models in dynamic context setting, we conduct test dialogues with the help of human users. During dynamic context evaluation, any minor change in user input may lead the dialogue down a completely different path. For this reason, we need to collect substantial amount test dialogues for evaluation. *Static context* setting mitigates this data collection challenge as the same contexts are used for evaluating different dialogue models. But it comes at the price of not accounting for the interactivity inherent in dialogue.

The test dialogues collected by both of these processes can then be used for evaluation. Sometimes these needs further analysis - which may include annotations, collecting subjective judgments from humans, etc. Since human judgments tend to vary, we may need to employ multiple judges. These are some of the reasons why evaluation performed by humans (as presented in Section 4.7) is time consuming, costly and sometimes prohibitively expensive.

Furthermore, if the system being developed contains a machine learning component, the problem of costly evaluation becomes even more serious. Machine learning components often optimize certain free parameters by using evaluation results on held-out data or by using n-fold cross-validation. Evaluation results can also help with feature selection. But if evaluation is time-consuming and costly, then the need for repeated evaluation can forbid the use of data-driven machine learning components.

For these reasons, using an automatic evaluation measure as an understudy is quickly becoming a common practice in natural language processing tasks. E.g., BLEU (Papineni et al., 2001) for machine translation, ROUGE (Lin and Hovy, 2003) for summarization. The general idea is to find an automatic evaluation metric that correlates very well with human judgments. This allows developers to use the automatic metric as a stand-in for human evaluation. Although it cannot replace the finesse of human evaluation, it can provide a crude idea of progress which can later be validated.

In this chapter we present automatic evaluation methods for evaluating dialogue models. We present two evaluation understudy measures for two different settings – static context and dynamic context. For automatic evaluation in static context setting, we follow a method similar to that used by DeVault et al. (2011b). For dynamic context setting, we propose an evaluation understudy based on the information ordering task.

Previously in section 2.3, we summarized the related work in evaluation of dialogue systems. Almost all of these methods have substantial human involvement for annotations, collecting subjective

judgments or filling in questionnaires for user satisfaction. This forms the motivation for finding a suitable evaluation understudy for dialogue models. In the next section we evaluate an automatic evaluation metric, *weak agreement*, originally proposed by (DeVault et al., 2011b) for static context evaluation. We evaluate how well it correlates with human judgments. We also introduce a new evaluation metric that outperforms *weak agreement* in terms of correlation with human judgments. Next, we introduce the task of information ordering for dynamic context evaluation of dialogue models. It is followed by the details of the experiments we carried out and our observations. We conclude with a summary and directions for future work.

5.1 Static Context Evaluation

In static context evaluation, a dialogue model is used to predict the most appropriate response for a given dialogue context. This dialogue context is not affected by the dialogue model being evaluated and the set of contexts used is held fixed. We are interested in automatic evaluation of dialogue models that operate at surface text level such as the models presented in section 4.6. We have collected data from four wizards as to which utterances are appropriate responses for given dialogue contexts (as described in section 4.7.2.1). We have also collected ratings for appropriateness of responses from different dialogue models on a scale of 1 to 5 (as described in section 4.7.2.2). These ratings were provided by four human judges for the same dialogues as used in wizard data collection. We have collected appropriateness ratings for a total of 397 unique pairs of $\langle u_t, context_t \rangle$, where u_t is a response utterance for a dialogue context $context_t$. We use this data for proposing and evaluating automatic evaluation measures in static context setting.

5.1.1 Weak Agreement

DeVault et al. (2011b) used an automatic evaluation measure based on wizard data collection for evaluating various dialogue models in a static context setting. The dialogue models evaluated in that study operate at the dialogue act level and consequently the wizard data collection is also done at the dialogue act level. Their proposed automatic evaluation, *weak agreement*, judges the response dialogue act for a given context as appropriate if any one of the wizards has chosen that dialogue act as an appropriate response. In their study DeVault et al. do not correlate this automatic measure with human judgments of appropriateness.

Let $R(u_t, context_t)$ denote the average appropriateness of the response utterance u_t for the dialogue context $context_t$ as judged by the four human judges. Also let $W(context_t)$ be the union of set of responses judged appropriate for the dialogue context $context_t$ by the four wizards. Then following DeVault et al. (2011b), an automatic evaluation for response appropriateness along the lines of *weak agreement* can be defined as,

$$R_{weak}(u_t, context_t) = \begin{cases} 5 & \text{if } u_t \in W(context_t) \quad \text{Appropriate response} \\ 1 & \text{if } u_t \notin W(context_t) \quad \text{Inappropriate response} \end{cases} \quad (5.1)$$

In order to test the validity of this automatic evaluation metric (R_{weak}), We correlate it with human judgments (R). This correlation can be computed either at the level of an individual response (i.e., for every unique value of $\langle u_t, context_t \rangle$) or at the system level (i.e., by aggregating the ratings over each dialogue model). The Pearson’s correlation between R_{weak} and R is 0.485 ($p < 0.001, n = 397$) at individual response level and 0.803 ($p < 0.05, n = 7$) at the system level. Thus, *Weak Agreement*, R_{weak} turns out to be a good evaluation understudy for judging appropriateness of responses given a dialogue context especially at the system level.

5.1.2 Voted Appropriateness

We made an observation regarding R_{weak} which may lead to an improvement. According to weak agreement, we should expect *Wizard Max Voted* and *Wizard Random* models to have the same appropriateness rating of value 5 (by definition in Equation 5.1). Instead, we observe that *Wizard Max Voted* model receives significantly higher appropriateness ratings than *Wizard Random*. This indicates that not all responses chosen by wizards are judged as highly appropriate by other judges. It also suggests that more votes from wizards for a response utterance are likely to result in higher appropriateness ratings.

Based on these observations, we propose an evaluation understudy *Voted Appropriateness*, R_{voted} . Let $V(u_t, context_t)$ be the number of wizards who chose the utterance u_t as an appropriate response to the dialogue context $context_t$. Following PARADISE (Walker et al., 2000), which models user satisfaction as a linear regression of observable dialogue features, we model R_{voted} as a linear regression based on V .

$$R_{voted}(u_t, context_t) = \alpha_0 + \alpha_1 \cdot V(u_t, context_t) \quad (5.2)$$

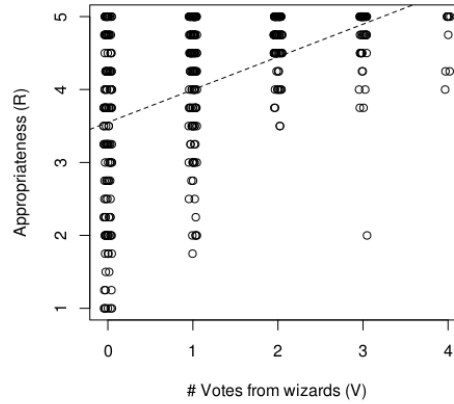


Figure 5.1: Appropriateness of responses (R) as judged by 4 human judges plotted against the number of wizard votes (V) received by those responses. The dashed line indicates a fitted linear model. A small amount of jitter is added to V for visualization.

Figure 5.1 shows the appropriateness rating (R) as judged by human judges for response utterances as a function of number of wizard votes (V) received by those response utterances. For this analysis we use only distinct pairs of $\langle u_t, context_t \rangle$ ($n = 397$). We fit a linear regression model for this data. The number of votes received V is a significant factor in estimating R ($p < 0.001$). The final linear model estimated from all available data is, $R_{voted} = 3.549 + 0.449V$. The fraction of variance explained by the model is 0.238.

To verify whether a simple linear regression model can be used as an automatic evaluation for static context setting, we perform 5-fold cross-validation analysis. During each fold, we hold out the data corresponding to one of the dialogues and train a linear model on the rest of the data. We use this trained model to compute *voted appropriateness* (R_{voted}) for the held-out data and then correlate it with the actual observed value of appropriateness rating (R) as judged by humans. The Pearson’s correlation between R_{voted} and R is 0.479 ($p < 0.001, n = 397$) at the individual response level. At the the system level the Pearson’s correlation between R_{voted} and R is 0.893 ($p < 0.01, n = 7$). At the system level, R_{voted} is a better evaluation understudy than R_{weak} . Figure 5.2 shows a comparison between these two possible evaluation measures for automatic evaluation of appropriateness in static context setting.

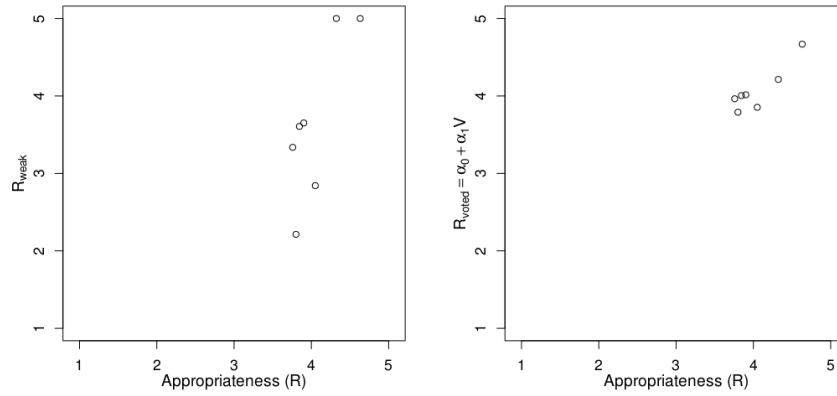


Figure 5.2: Comparison between two automatic evaluation understudy measures at system level in static context setting.

5.1.3 Discussion

Different resources are required to build different automatic evaluation measures. For R_{weak} , we need to collect wizard data as described in section 4.7.2.1. When this data is being collected at the surface text level, we need a substantial number of wizards (four or more) each selecting a large number of appropriate responses for each context. For the automatic evaluation measure R_{voted} , in addition to the wizard data we need resources to estimate the linear regression model. As training data to build a linear regression model, we need human evaluators' appropriateness ratings for responses given the dialogue contexts.

Automatic evaluation for static context setting involves human efforts for collecting wizard data and appropriateness ratings. But since the resources are collected at the surface text level, with the use of appropriate tools (such as shown in Figure 4.21) *non-experts* can accomplish this task. Moreover since static context setting uses a fixed set of contexts, wizard data collection needs to be performed only once. The resulting automatic evaluation metrics can be used to compare different dialogue models.

When using the *Voted Appropriateness* evaluation method, the training data used for linear regression should represent all possible responses adequately. The data used to fit our model does not include lower baseline models like *Random* and *Segmented Random*. This results in a rather high intercept value of 3.549. For any model producing responses that are not judged appropriate by any of the wizards, our model would predict the appropriateness value of 3.549 which seems rather high.

5.2 Dynamic Context Evaluation

In dynamic context setting, the dialogue context itself is dependent on the dialogue model being evaluated. If we were to use an approach similar to static context setting, then we would need to collect wizard data for all possible contexts that can be seen during a dynamic context setting. This number of possible contexts is exponentially large and it is impractical to collect wizard data for it. Another possible approach towards automatic evaluation in dynamic context setting is to use a simulated user and conduct simulated dialogue runs. Then for an automatic evaluation we need a way to evaluate the simulated dialogues (generally done in terms of accumulated reward). Also building a simulated user can get as complex as building a dialogue system and is generally done at the dialogue act level.

A task such as information ordering (described in Section 5.2.1) has the desired property where the context used in predicting the next utterance is itself dependent on the model being evaluated. Recently, the discourse coherence modeling community has started using the information ordering task as a testbed for discourse coherence models (Barzilay and Lapata, 2005; Soricut and Marcu, 2006). Lapata (2006) has proposed an automatic evaluation measure for this information ordering task. Here we propose to use the same task as a testbed for evaluating dialogue coherence models. The discourse coherence models are designed for non-interactive forms of communication such as news articles, written documents. Whereas, dialogue is a result of an interactive process. It is not clear whether evaluation using information ordering can be applied to dialogue models without any modifications. Here we evaluate the viability of the information ordering task for evaluating dialogue models. We also propose an evaluation understudy for dialogue coherence models in dynamic context setting.

5.2.1 Information Ordering

The information ordering task consists of choosing a presentation sequence for a set of information bearing elements. When the models are operating at the surface text level, these information-bearing elements are sentences or utterances rather than high-level concepts or dialogue acts. The information ordering task is well suited for text-to-text generation such as single or multi-document summarization (Barzilay et al., 2002). Recently there has been a lot of work in discourse coherence modeling (Lapata, 2003; Barzilay and Lapata, 2005; Soricut and Marcu, 2006) that has used information ordering to test the coherence models.

One of the ways to use the information ordering task for evaluating discourse models is to start with a known presentation sequence. We assume that a human authored document is an example of coherent discourse and the ordering of sentences in this document forms a reference ordering. Next we use the discourse model to recreate a presentation order for this set of sentences that maximizes the coherence of the presentation as judged by the discourse model. This presentation order can then be compared with the reference order. A discourse model which is better at judging coherence will have a presentation order close to the reference order. There are various ways in which one can measure the closeness between a presentation order and a reference order. Lapata (2003) proposed Kendall’s τ , a rank correlation measure, as one such candidate.

We propose to use the information ordering task to test dialogue models. In our case for evaluating dialogue models, we begin with a human-human dialogue as an example of a coherent dialogue. We split this dialogue into individual information bearing units. Next we use the dialogue model to recreate a presentation order for these units and evaluate the closeness between the most coherent presentation order generated by the dialogue model and the reference order.

A dialogue can be divided into units at different levels of granularity. For this study, the information bearing units will be dialogue turns. A dialogue turn, where one speaker holds the floor can be thought of as composed of one or more utterances. The information bearing units can also be at the utterance level, but we choose to remain consistent with our earlier approach of retrieving a whole turn that was used by dialogue models described in chapter 4. Figure 5.3 shows an example of a random permutation of a human-human dialogue and the task is to recover the original reference order.

There are a few elements common between evaluating a dialogue model in a dynamic context setting and the information ordering task. First, both these tasks employ the *selection* approach i.e. selecting the most appropriate or coherent next utterance given a dialogue context. Second, in both these tasks the dialogue context itself is dependent on the previous choices made by the dialogue model. There is one difference however, when evaluating a dialogue model in a dynamic context setting (as described in Section 4.7.1) the model is always selecting from the full set of possible responses (U_{train}). But

1	User	that's the one
2	Agent	ok and what do you need to do?
3	User	ok on June sixth from San Jose to Denver, United
4	Agent	leaving at what time?
5	User	ok
6	Agent	AAA at American Express may I help you?
7	User	I believe there's one leaving at eleven o'clock in the morning
8	Agent	leaves at eleven a.m. and arrives Denver at two twenty p.m. out of San Jose
9	User	yeah this is BBB BBB I need to make some travel arrangements
10	Agent	yeah that's United flight four seventy

Figure 5.3: A random permutation of a human-human dialogue [Source: (Bratt et al., 1995)]. The original reference order can be recovered by reading the dialogue in sequence $\langle 6, 9, 2, 3, 4, 7, 8, 5, 10, 1 \rangle$.

for information ordering task the set of utterances to choose from is decreasing with each decision as repetitions are not allowed in the presentation order.

As seen in Section 2.2, Purandare and Litman (2008) only allow for a binary classification of sequences as either coherent (original reference order) or incoherent (any other permutation). For comparing different dialogue coherence models, we need the ability for finer distinction between sequences of information elements being put together. For the discourse setting, Lapata (2006) have proposed and evaluated Kendall's τ , a rank correlation measure, as one such candidate. They show that human judges can reliably provide coherence ratings for various permutations of text. (Pearson's correlation for inter-rater agreement is 0.56) and that Kendall's τ is a good indicator for human judgment for coherence (Pearson's correlation for Kendall's τ with human judgment is 0.45 ($p < 0.01$)).

There are certain advantages offered by using information ordering as a task to evaluate dialogue coherence models. First, the task does not require a human subject to take part in conversations in an interactive manner. Second, the task is agnostic about the underlying dialogue model. It can be a data-driven statistical model or information-state based, form based or even a reinforcement learning system based on MDP or POMDP. Third, there are simple objective measures available to evaluate the success of information ordering task.

Before adapting the information ordering task for dialogues, certain questions need to be answered. We need to validate that humans can reliably perform the task of information ordering for dialogues and can judge the coherence for different sequences of dialogue turns. We also need to find which objective measures (like Kendall's τ) correlate well with human judgments.

5.2.2 Evaluating Information Ordering

One of the advantages of using information ordering as a testbed is that there are objective measures available to evaluate the performance of the information ordering task. These objective measures judge the closeness of an observed presentation sequence with the reference sequence. Kendall's τ (Kendall, 1938), a rank correlation coefficient, is one such measure. Given a reference sequence of length n , Kendall's τ for an observed presentation sequence can be defined as,

$$\tau = \frac{\# \text{ concordant pairs} - \# \text{ discordant pairs}}{\# \text{ total pairs}}$$

Each pair of elements in the observed presentation sequence is marked either as concordant - appearing in the same order as in the reference sequence or as discordant otherwise. The total number of pairs

is $C_2^n = n(n-1)/2$. τ ranges from -1 to 1. Kendall's τ can be thought of as measuring global features of coherence.

Another possible measure can be defined as the fraction of n-grams from the reference sequence that are preserved in the observed sequence.

$$b_n = \frac{\# \text{ n-grams preserved}}{\# \text{ total n-grams}}$$

In this study we have used, b_2 , fraction of bigrams and b_3 , fraction of trigrams preserved from the reference sequence. These values range from 0 to 1. The measure b_n can be thought of as evaluating local features of coherence.

Table 5.1 gives examples of observed sequences and respective b_2 , b_3 and τ values. If the observed sequence is same as reference sequence then $b_2 = b_3 = \tau = 1$ (the highest values). If it exactly reverse then, $b_2 = b_3 = 0$ and $\tau = -1$ (the lowest values). The third row in the table represents an ordering where the last two turns are presented first and the rest of the sequence is same as the reference. This may represent a situation where utterances 8 and 9 form an adjacency pair and that exchange takes place before the rest of the dialogue. Kendall's τ punishes such sequences very heavily as compared to b_2 or b_3 . Rows 4 & 5 from Table 5.1 show observed sequences where not a single local feature of coherence is preserved ($b_2 = b_3 = 0$) but Kendall's τ values swing from a high 0.60 to a low -0.64. In these examples we can notice how τ accounts for long-distance relationships whereas b_2 , b_3 are sensitive to local features only.

Observed Sequence	b_2	b_3	τ
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]	1.00	1.00	1.00
[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]	0.00	0.00	-1.00
[8, 9, 0, 1, 2, 3, 4, 5, 6, 7]	0.89	0.75	0.29
[4, 1, 0, 3, 2, 5, 8, 7, 6, 9]	0.00	0.00	0.60
[6, 9, 8, 5, 4, 7, 0, 3, 2, 1]	0.00	0.00	-0.64
[2, 3, 0, 1, 4, 5, 8, 9, 6, 7]	0.56	0.00	0.64

Table 5.1: Examples of observed sequences and their respective b_2 , b_3 & τ values. Here the reference sequence is [0,1,2,3,4,5,6,7,8,9].

5.2.3 Experimental Setup

Following Lapata (2006), we set up experiments to find an objective metric that correlates well with human judgments for the information ordering task. For our experiments we used segments drawn from 9 dialogues. These dialogues were two-party human-human dialogues. To ensure applicability of our results over different genres and domains of dialogue, we chose these 9 dialogues from different sources. Three of these were excerpts from SASO-ST (Traum et al., 2005) role-play dialogues involving negotiations from the same corpus used as a testbed for developing models presented in Chapter 4 (See Figure 5.4 for an example). Three are from SRI's Amex Travel Agent data which are task-oriented dialogues about air travel planning (Bratt et al., 1995) (See Figure 5.5 for an example). The rest of the dialogues are scripts from popular television shows (See Figure 5.6 for an example). Each excerpt drawn was 10 turns long with turns strictly alternating between the two speakers.

Following the experimental design of (Lapata, 2006) we created random permutations for these dialogue segments. We constrained our permutations so that the permutations always start with the same

Doctor	hello i'm doctor perez how can i help you
Captain	uh well i'm with uh the local i'm i'm the commander of the local company and uh i'd like to talk to you about some options you have for relocating your clinic
Doctor	uh we're not uh planning to relocate the clinic captain what uh what is this about
Captain	well have you noticed that there's been an awful lot of fighting in the area recently
Doctor	yes yes i have we're very busy we've had many more casual+ casualties many more patients than than uh usual in the last month but uh what what is this about relocating our clinic have have uh you been instructed to move us
Captain	no but uh we just have some concerns about the increase in fighting xx
Doctor	are you suggesting that we relocate the clinic because we had no plans we uh we uh we're located here and we've been uh we are located where the patients need us
Captain	yeah but yeah actually it is a suggestion that you would be a lot safer if you moved away from this area we can put you in an area where there's n+ no insurgents and we have the area completely under control with our troops
Doctor	i see captain is this a is this a suggestion from your commander
Captain	i'm uh the company commander

Figure 5.4: A segment from a negotiation role-play dialogue.

Agent	American Express may I help you?
User	oh A this is B
Agent	ok wh- how can I help you?
User	we have an emergency traveler who is going t- uh Copenhagen from San Francisco to Los Angeles and he is going on a two o'clock flight.
Agent	he's going to Copenhagen today?
User	that's correct yes
Agent	it's just a flat one way?
User	ah yes that's correct
Agent	in business class?
User	ok

Figure 5.5: A segment from a travel agent dialogue.

speaker as the original dialogue and turns strictly alternate between the speakers. With these constraints there are still $5! \times 5! = 14400$ possible permutations per dialogue. We selected 3 random permutations for each of the 9 dialogues. In all, we have a total of 27 dialogue permutations. They are arranged in

AJ	Lieutenant, how's your memory of the Gulf War?
Bud	Foggy, sir. Uh, given all that's happened in the region since.
AJ	You recall a missing Petty Officer by the name of Allison La Porte?
Bud	No, sir.
AJ	Fell out of a medevac helo over southern Iraq. Was 22 at the time. A hot LZ, couldn't find her until yesterday, when she was caught attempting to steal antibiotics from a corpsman in the village of, uh Al-Muntassir.
Bud	That's remarkable, sir. Did she explain how she was able to survive for 12 years?
AJ	As a Bedouin. Joined a nomadic tribe and married its sheik.
Bud	Willingly, sir?
AJ	Well, that's for you to find out. I'm authorizing a JAGMAN investigation to Camp Babylon in southern Iraq.
Bud	You're sending me, sir?

Figure 5.6: A segment from a television show dialogue. [source: <http://www.twiztv.com/>]

3 sets, each set containing a permutation for all 9 dialogues. We ensured that not all permutations in a given set are particularly very good or very bad. We used Kendall's τ to balance the permutations across the given set as well as across the given dialogue.

Unlike Lapata (2006) who chose to remove the pronouns and discourse connectives, we decided not do any pre-processing on the text like removing disfluencies or removing cohesive devices such as anaphora, ellipsis, discourse connectives, etc. One of the reason is that such pre-processing, if done manually, defeats the purpose of removing humans from the evaluation procedure. Moreover it is very difficult to remove certain cohesive devices such as discourse deixis without affecting the coherence level of the original dialogues.

5.2.4 Viability of Information Ordering Task and Human-level Upper Baseline

We want to know whether information ordering task is a viable approach for evaluating dialogue models. Within this evaluation framework, the performance of a dialogue model is judged by the closeness of the model-generated presentation order and the original reference order. There is an assumption here which says that there is a unique coherent ordering possible which is same as the original reference order. If there is a possible coherent ordering that is different from the reference order and a dialogue model were to generate such a coherent ordering, then evaluation measures that judge the closeness from the original reference order (such as Kendall's τ , b_2 , b_3 introduced earlier) may underestimate the performance of the dialogue model. We set up an experiment to see how often humans come up with a coherent ordering that is different from the original reference order and how different it is. In this experiment we also seek to establish a higher baseline for the task of information ordering in dialogues.

We presented the dialogue permutations to our human judges and asked them to reorder the turns so that the resulting order is as coherent as possible. A total of 11 judges participated in this experiment. They were presented with a drag and drop interface over the web that allowed them to reorder the dialogue permutations. The reordering was constrained to keep the first speaker of the reordering the same as that of the original reference dialogue and the re-orderings must have strictly alternating turns. Each human judge was presented with 9 permutations, one for each original dialogue. This resulted in a total of $11 \times 9 = 99$ reordered dialogue permutations.

We computed Kendall's τ and the average of fraction of bigrams and trigrams preserved, $(b_2 + b_3) / 2$, for these re-orderings. Figure 5.7a and 5.7b shows the frequency distribution of τ and $(b_2 + b_3) / 2$ values respectively. These figures show that for a majority of times, the human-generated coherent orderings were same as or very close to the original reference orderings. This proves the viability of the information ordering task and of the evaluation measures that are based on judging the closeness of orderings based on the original reference.

As an upper baseline for the information ordering task, humans achieve high performance. For Kendall's τ , the mean of the reordered dialogues is 0.82 (std dev = 0.25) and for $(b_2 + b_3) / 2$, the mean is 0.71 (std dev = 0.28). These values establish an upper baseline for the information ordering task. These can be compared against the random baseline. For τ random performance is 0.02¹ and for $(b_2 + b_3) / 2$ it is 0.11².

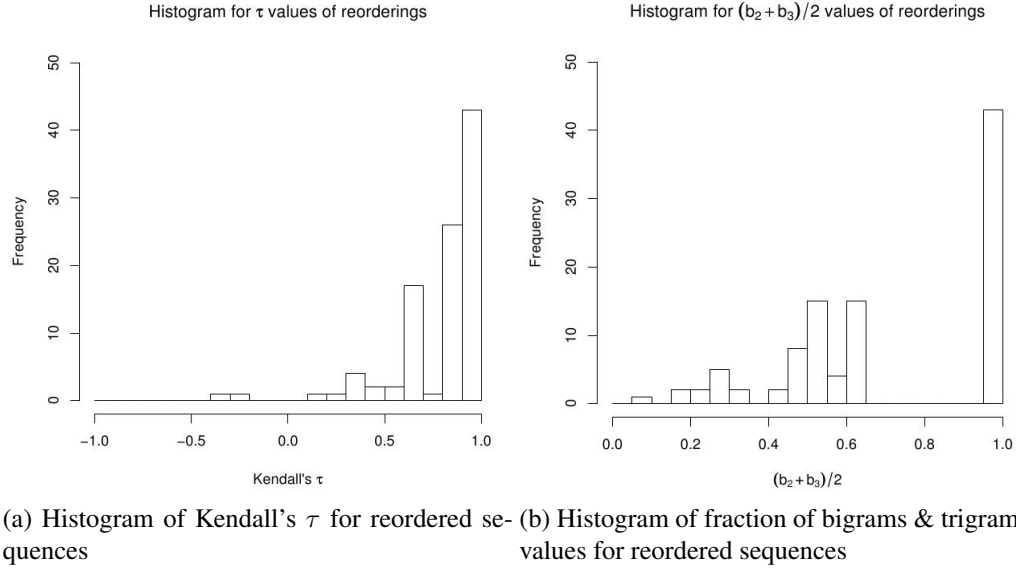


Figure 5.7: Human-level upper baseline for information ordering task (human performance)

5.2.5 Automatic Evaluation for the Information Ordering Task

In order to find out which evaluation measures are appropriate for evaluating information ordering task, we propose to collect human judgments of coherence for dialogue permutations and correlate these with possible automatic evaluation measures.

5.2.5.1 Holistic Evaluation

In this experiment a total of 9 human judges participated. They were divided among the 3 sets (3 judges per set). Each judge was presented with 9 dialogue permutations. They were asked to assign a single

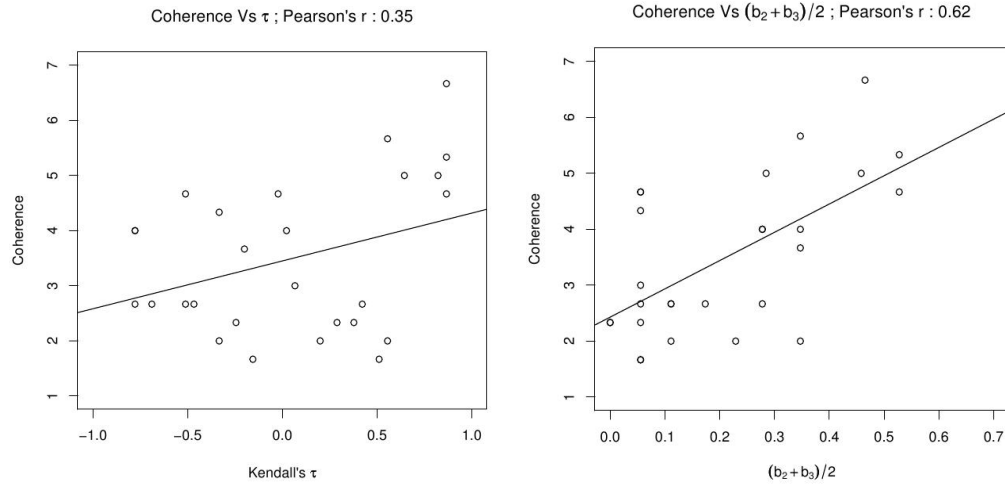
¹Theoretically this should be zero. The slight positive bias is the result of the constraints imposed on the re-orderings - like only allowing the permutations that have the correct starting speaker.

²This value is calculated by considering all 14400 permutations as equally likely.

coherence rating for each dialogue permutation. The ratings were on a scale of 1 to 7, with 1 being very incoherent and 7 being perfectly coherent. We did not provide any additional instructions or examples of scale as we wanted to capture the intuitive idea of coherence from our judges. Within each set the dialogue permutations were presented in random order.

We compute the inter-rater agreement by using Pearson's correlation analysis. We correlate the ratings given by each judge with the average ratings given by the judges who were assigned the same set. For inter-rater agreement we report the average of 9 such correlations which is 0.73 (std dev = 0.07). Artstein and Poesio (2008) have argued that Krippendorff's α (Krippendorff, 2004) can be used for inter-rater agreement with interval scales like the one we have. In our case for the three sets, α values were 0.49, 0.58, 0.64. These moderate values of alpha indicate that the task of judging coherence is indeed a difficult task, especially when detailed instructions or examples of scales are not given.

In order to assess whether Kendall's τ can be used as an automatic measure of dialogue coherence, we perform a correlation analysis of τ values against the average ratings by human judges. The Pearson's correlation coefficient is 0.35 and it is statistically not significant ($P=0.07$). Fig 5.8a shows the relationship between coherence judgments and τ values. This experiment fails to support the suitability of Kendall's τ as an evaluation understudy at least with the amount of data we collected.



(a) Kendall's τ does not correlate well with human judgments for dialogue coherence. (b) Fraction of bigram & trigram counts correlate well with human judgments for dialogue coherence.

Figure 5.8: Single coherence rating per permutation.

We also analyzed the correlation of human judgments against simple n-gram statistics, specifically $(b_2 + b_3)/2$. Fig 5.8b shows the relationship between human judgments and the average of fraction of bigrams and fraction of trigrams that were preserved in the permutation. The Pearson's correlation coefficient is 0.62 and it is statistically significant ($P<0.01$).

5.2.5.2 Turn-by-Turn Evaluation

Since human judges found it relatively hard to assign a single rating to a dialogue permutation, we decided to repeat the collection of human judgments for coherence. In this experiment we asked the

judges to provide coherence ratings at every turn, based on the dialogue context that preceded the turn. The dialogue permutations were presented to the judges through a web interface in an incremental fashion turn by turn as they rated each turn for coherence. See Figure 5.9 for the screenshot of the interface. We used a scale from 1 to 5 – with 1 being completely incoherent and 5 as perfectly coherent³. A total of 11 judges participated in this experiment with the first set being judged by 5 judges and the remaining two sets by 3 judges each.

Speaker	Text	Coherence Rating
JACK	Yeah this is Bauer.	
TONY	Jack, this is Tony. You guys are around 10 minutes from the airport?	★★★★☆
JACK	Do everything you have to.	★☆☆☆☆
TONY	No, not yet. We're still working on it. Look- you'll link up with the assault team, they're set outside the emergency room. They're sweeping the ground now, they're starting to lock it down.	★★★★☆
JACK	Okay, that sounds alright. Tony, contacted Kim yet?	★★★★☆
TONY	No, I haven't. I just called the sheriff's station, they sent out search teams. I just don't understand why she hasn't tried contacted us yet.	★★★★★
JACK	Yeah, that's seems about right. Look- you got a complete tail number on the aircraft yet?	★★★☆☆
TONY	Believe me; I'm putting every resource we can spare on it.	★★★★☆
JACK	She knows she can't go back to Los Angeles, and right now the LAPD's looking for her as a murder suspect. She's not gonna reach out to us- do you understand? We've got to find her. We've got to find her.	★★★★☆
TONY	Okay.	★★★★☆

Figure 5.9: Screenshot of the interface used for collecting coherence rating for dialogue permutations.

For the rest of the analysis, we use the average coherence rating from all turns as a coherence rating for the dialogue permutation. We performed the inter-rater agreement analysis as in experiment 1. The average of 11 correlations is 0.83 (std dev = 0.09). Although the correlation has improved, Krippendorff's α values for the three sets are 0.49, 0.35, 0.63. This shows that coherence rating is still a hard task even when judged turn by turn.

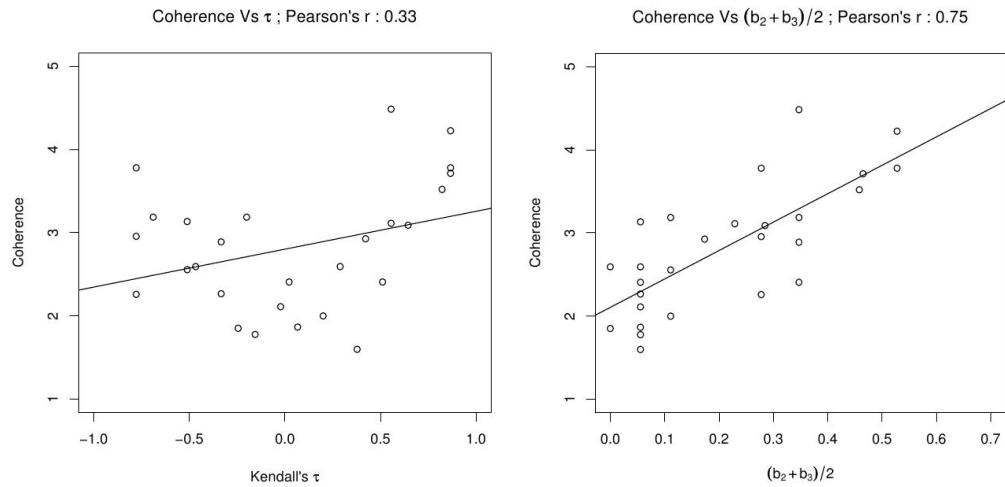
We assessed the relationship between the average coherence rating for dialogue permutations with Kendall's τ (see Fig 5.10a). The Pearson's correlation coefficient is 0.33 and is statistically not significant ($P=0.09$) with the data we have collected.

Figure 5.10b shows high correlation of average coherence ratings with the fraction of bigrams and trigrams that were preserved in permutation. The Pearson's correlation coefficient is 0.75 and is statistically significant ($P<0.01$). Results of both experiments suggest that, $(b_2 + b_3)/2$ correlates very well with human judgments and can be used for evaluating information ordering when applied to dialogues.

5.2.6 Discussion

Results show that $(b_2 + b_3)/2$ correlates with human judgments for dialogue coherence better than Kendall's τ . τ encodes long distance relationships in orderings where as $(b_2 + b_3)/2$ only looks at local context. Figure 5.11 shows the relationship between these two measures. Notice that most of the orderings have τ values around zero (i.e., in the middle range for τ), whereas the majority of orderings

³We believe this is a less complex task than Holistic evaluation experiment and hence a narrower scale is used.



(a) Kendall's τ does not correlate well with human judgments for dialogue coherence. (b) Fraction of bigram & trigram counts correlate well with human judgments for dialogue coherence.

Figure 5.10: Turn-by-turn coherence rating per permutation

will have a low value for $(b_2 + b_3)/2$. τ seems to overestimate coherence even in the absence of immediate local coherence (See the fourth entry in table 5.1). It seems that local context is more important for dialogues than for discourse, which may follow from the fact that dialogues are produced by two speakers who must react to each other, while discourse can be planned by one speaker from the beginning. Traum and Allen (1994) point out that such social obligations to respond and address the contributions of the other should be an important factor in building dialogue systems.

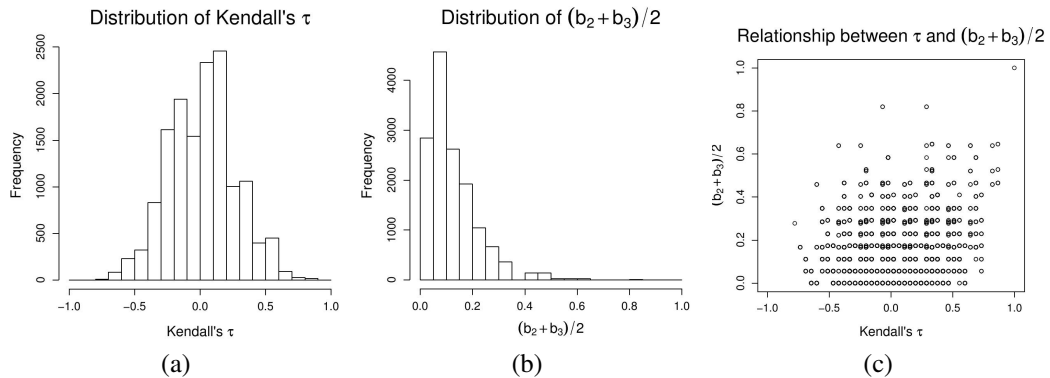


Figure 5.11: Distributions for Kendall's τ , $(b_2 + b_3)/2$ and the relationship between them for all possible dialogue permutations with 10 turns and earlier mentioned constraints.

The information ordering paradigm does not take into account the content of the information-bearing items. E.g., Turns like "yes", "I agree", "okay" perform the same function and should be treated as

exchangeable. This may suggest a need for modification to the objective metrics used to evaluate the information ordering task when dialogues contain such utterances.

Human judges can find the optimal sequences with relatively high frequency, at least for short dialogues. It remains to be seen how this varies with longer dialogues which may contain sub-dialogues that can be arranged independently of each other.

As with any evaluation understudy, one must be careful while using it as the correlation with human judgments is not perfect and may be inaccurate in some cases. It can not completely replace the need for full evaluation with human judges in all cases (see [Callison-Burch et al. \(2006\)](#) for a critique of BLUE along these lines). Despite this we believe an evaluation understudy will prove useful in solving the evaluation bottleneck to some extent.

5.3 Conclusion

One of the goals of this thesis is rapid and cost-efficient evaluation of dialogue models. But evaluating a dialogue model involves a lot of human effort and repetitive evaluations of dialogue model can be prohibitively costly. Our approach towards reducing the cost of estimating the performance of a dialogue model is to reduce the human involvement in the evaluation process. Towards that end, in this chapter we have proposed automatic evaluation measures for dialogue models in two different evaluation settings – *Static context* and *Dynamic context*. To judge the effectiveness of these automatic evaluation measures we have correlated them with human judgments.

For static context, we need to collect wizard data which can be accomplished by *non-experts* with the help specialized wizard tools. In this chapter, for the first time, we have evaluated *Weak Agreement*, which correlates well with human judgments (Pearson’s $r = 0.80$) at the system level. We also proposed an improvement which leads to another evaluation understudy, *Voted Appropriateness*, which correlates better with human judgments (Pearson’s $r = 0.89$) at the system level.

For dynamic context, we propose to use the task of information ordering to evaluate dialogue models. We establish the viability of this task for dialogues and propose an evaluation understudy based on the fraction of n-grams preserved, $(b_2 + b_3)/2$. We correlate this evaluation understudy with human judgments (Pearson’s $r = 0.75$) and show it outperforms Kendall’s τ which has been found to be suitable for discourse.

Using these automatic evaluation measures that correlate well with human judgments will lead to mitigating the evaluation bottleneck.

Chapter 6

Conclusion

6.1 Summary

The goal of this thesis is to enable cost-effective and rapid prototyping and evaluation of dialogue systems. We summarize the work presented here by repeating the thesis statement in parts and showing how the specific thesis contributions support it.

In case of dialogue-act based architecture, once the type of resources to be used have been determined, the cost of developing a dialogue system can be lowered by reducing the cost of building those specific resources. Our approach to reducing this cost is to allow *non-experts* to build such resources with the help of *integrated authoring tools*.

In chapter 3, we have explained how advanced question answering virtual humans can be rapidly developed by *non-experts*. The advanced dialogue behaviors expected from these characters dictate the use of an information-state based dialogue manager, which in turn requires a specific set of resources. These resources include the domain knowledge of the character, all the relevant dialogue acts for this domain and examples of surface text for these dialogue acts. The *genre-specific* integrated authoring tool, DomainEditor provides a simple ontology for defining the domain. The tool uses a *genre-specific minimalist* dialogue act schema and hides the complexity involved in authoring the dialogue acts, which allows *non-experts* to collect the required resources *consistently* and *completely*. The tool supports both *top-down* and *bottom-up* phases of character development. To date, 10 virtual humans have been developed by *non-experts* using this authoring tool and the development time can be as low as two weeks. In chapter 3, we have shown that by using a *genre-specific* dialogue manager and an integrated authoring tool, *non-experts* can rapidly build advanced question answering characters.

For surface-text based architectures, we posit that for certain virtual human conversational domains, a response to a dialogue context can be formulated by simply selecting an appropriate response from a set of utterances from an in-domain human-human dialogue corpus.

In chapter 4, we compared two approaches to response formulation in dialogue models: *Generation* and *Selection*. We have presented a method to verify the viability of the selection approach for given domain and available corpus. We find that some virtual human domains are amenable to the selection approach where over half of the utterances have been seen before in the training corpus.

We develop flexible architectures that allow novel combinations of different types of resources, such as surface text transcripts and information state annotations. This allows us

to better understand the relative utility of different resources and the resulting information can be used to optimize the cost and/or performance of the dialogue system.

In chapter 4, we presented unsupervised dialogue models that work primarily at the surface text level and employ the *selection* approach for formulating a response. The virtual human dialogue system based on such models can be bootstrapped from only the resource of un-annotated in-domain human-human corpus. Resources such as information state annotations for topics, etc. can be incrementally added to the dialogue model as they become available. We have evaluated these models in different settings – *dynamic context* or *static context*, and with different evaluator choices – dialogue participants themselves or bystanders, and with different modalities – typed text or speech. In dynamic context evaluation by bystanders, the best performing model *Text Segmented Nearest Context* has achieved upto 68% of human-level performance above the *Text Random* baseline. In static context evaluation, the advanced text-to-text models which use the utterance content information in addition to context information perform better than the simpler *Nearest Context* model which uses context information alone. We have also shown that the *Perceptron* model easily allows novel combinations of resources by simply changing the set of features used which allows us to understand relative utility of different resources.

In order to explore a wide variety of resource combinations, we need to reduce the cost of evaluating the performance of a dialogue system. We develop automatic evaluation measures, which do not require human participation, and correlate well with human judgments.

In chapter 5, we have proposed automatic evaluation measures for dialogue models in two different evaluation settings – *Static context* and *Dynamic context*. To judge the effectiveness of these automatic evaluation measures we have correlated them with human judgments. For static context, we have evaluated *Weak Agreement*, which correlates well with human judgments (Pearson’s $r = 0.80$) at the system level. We also proposed an improvement which leads to another evaluation understudy, *Voted Appropriateness*, which correlates better with human judgments (Pearson’s $r = 0.89$) at the system level. For dynamic context, we propose to use the task of information ordering to evaluate dialogue models. We have established the viability of this task for dialogues and proposed an evaluation understudy based on the fraction of n-grams preserved, $(b_2 + b_3)/2$. We have correlated this evaluation understudy with human judgments (Pearson’s $r = 0.75$) and show it outperforms Kendall’s τ which has been found to be suitable for discourse. Using these automatic evaluation measures that correlate well with human judgments will lead to mitigating the evaluation bottleneck.

6.2 Future Work

There are a number of avenues for future work that are related to but not part of this dissertation.

Extensions to Integrated Authoring Tool and Dialogue Manager

In chapter 3, we presented an approach of reducing the cost of building dialogue systems by employing *non-experts* to build virtual human dialogue systems. The approach relied on *genre-specific* authoring tools and dialogue manager. The *minimalist* dialogue act schema and the dialogue management state machines are still authored by experts. One possible future direction is to extend the authoring tool and dialogue manager to handle other genres of dialogue. The work presented in chapter 3 began with the genre of simple tactical questioning and was later generalized to all types advanced question answering dialogues (e.g., Virtual humans, Amber and Victor, that could lie; and Bradley who could use humor). Recently, we’ve also experimented with build simple negotiation characters (e.g., Jabbar and Sadik). But

still most of the virtual human's dialogue behavior is reactive and the initiative mostly remains with the user. A separate mechanism for virtual humans to take initiative would be an ideal addition to dialogue manager. This would require corresponding additions to the dialogue act schema to handle new types of dialogue acts and to the authoring tool for specifying when and what initiative to take. Ability to take initiative in a dialogue would allow our approach to incorporate more genres of dialogue.

Dialogue Management as Sequence Planning

The dialogue models we presented in chapter 4 simplify the dialogue management by assuming that the task is to simply predict the most appropriate response given a dialogue context. But dialogue has also been modeled as a sequential decision making process. Models such as Markov Decision Process and Partially Observable Markov Decision Process have been used for dialogue modeling (Levin et al., 2000; Williams and Young, 2007). One of the criticism for such models has been that the specification of the reward function is generally guided by intuition and the reward function manually tuned to get the appropriate behavior. The learned strategy depends on this reward function. For some dialogue systems, such as those used for virtual humans where the goal is to be as human like as possible, designing a reward function is very difficult. Standard dialogue features such as efficiency in term of dialogue length and task completion cannot be used. Instead it makes sense to have a strategy that imitates the training corpus (which are human-human dialogues) as closely as possible in order to be most believable, as was done for the models presented in chapter 4. But instead of producing the most appropriate utterance based on an observed context, a dialogue model can be modified to look-ahead a few steps and choose the response that maximizes the expected coherence over the next few turns. At certain points in dialogue it may be better to not simply react to the user contribution, but instead take an initiative and drive the dialogue along a more familiar trajectory.

Evaluation Understudy as Feedback

We presented automatic evaluation measures in chapter 5 for different evaluation settings. These evaluation measures are designed to be used to compare different dialogue models. But these can also be used as a feedback in improving a dialogue model especially the dialogue models that use machine learning techniques. Such models generally follow iterative process for improvement and require repetitive evaluation. There are many parameters that require tuning for such data-driven models. Automatic evaluation measures can be used for estimating these parameters in a fashion similar to Och (2003).

Coherence Vs Cohesion

In our work on evaluation understudy using the information ordering task, we did not pre-process the utterances when we made up random reorderings of the dialogue. One possible thing would be to avoid the improper use of cohesive devices that results from reordering. Halliday and Hasan (1976) argue that cohesive devices used in a text distinguish it as a coherent piece of text from just a collection of word tokens. These cohesive devices include reference, substitution, ellipsis and lexical relationships. Lapata (2006) replace the pronouns that cannot be resolved within the sentence with its referents and disallow use of discourse connectives. One can repeat our experiments with such modifications to tease apart the effects of cohesive devices. We chose not to pre-process the utterances for several reasons. First of all, this does introduce more pre-processing by humans, precisely what we are trying to avoid. Secondly, it is not clear how to remove all kinds of cohesive devices like ellipsis. Furthermore, Brown and Yule (1983) argue that the cohesive devices are not the primary reason for coherence of the text. Instead, it is the semantic relations between the text elements that are responsible for coherence. They argue that

cohesive devices alone will not be of much help in re-constructing the text from random permutations. Still it would be interesting to perform the experiments described in section [5.2.4](#) and [5.2.5](#), where the information bearing units (turns) would not have improper use of cohesive devices. One possibility would be to use a complete dialogue act specification rather than surface text for turns.

Bibliography

- Abu Shawar, B. and Atwell, E. (2003). Using dialogue to retrain a chatbot system. In *In proceedings of corpus linguistics*. 7
- Abu Shawar, B. and Atwell, E. (2005). Using corpora in machine-learning chatbot systems. *International Journal of Corpus Linguistics*, 10:489–516.
- Alexandersson, J. (1996). Some ideas for the automatic acquisition of dialogue structure. In *Proceedings of the Eleventh Twente Workshop on Language Technology (TWLT 11): Dialogue Management in Natural Language Systems*, pages 149–158. 13, 23
- Allen, J. F., Schubert, L. K., Ferguson, G., Heeman, P., Hwang, C. H., Kato, T., Light, M., Martin, N. G., Miller, B. W., Poesio, M., and Traum, D. R. (1995). The trains project: A case study in building a conversational planning agent. *Journal of Experimental and Theoretical AI*, 7:7–48. 3, 4
- Alshawi, H. and Douglas, S. (2001). Variant transduction: a method for rapid development of interactive spoken interfaces. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*, pages 1–9, Morristown, NJ, USA. Association for Computational Linguistics. 24
- Army (2006a). Human intelligence collector operations. Technical Report FM 2-22.3, Department of the Army. Appendix H: SALUTE Reporting. 44
- Army (2006b). Police intelligence operations. Technical Report FM 3-19.50, Department of the Army. Appendix D: Tactical Questioning. 28
- Artstein, R., Gandhe, S., Gerten, J., Leuski, A., and Traum, D. (2009a). Semi-formal evaluation of conversational characters. In *Languages: From Formal to Natural. Essays Dedicated to Nissim Francez on the Occasion of His 65th Birthday*, volume 5533 of *Lecture Notes in Computer Science*. Springer. 58
- Artstein, R., Gandhe, S., Leuski, A., and Traum, D. (2008). Field testing of an interactive question-answering character. In *proceedings of the ELRA Workshop on Evaluation, LREC*. 2, 5, 7, 28, 59, 86
- Artstein, R., Gandhe, S., Rushforth, M., and Traum, D. (2009b). Viability of a simple dialogue act scheme for a tactical questioning dialogue system. In *DiaHolmia, 13th Workshop on the Semantics and Pragmatics of Dialogue*, Stockholm, Sweden. 17, 42, 43, 44
- Artstein, R. and Poesio, M. (2008). Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596. 43, 99
- Artstein, R., Rushforth, M., Gandhe, S., Traum, D., and Donigian, A. (2011). Limits of simple dialogue acts for tactical questioning dialogues. In *Proceedings of 7th IJCAI workshop on Knowledge and Reasoning in Practical Dialogue Systems*, Barcelona, Catalonia (Spain). 17, 42, 43, 44

- Austin, J. L. (1962). *How to Do Things With Words*. Harvard University Press. 5
- Aylett, M. P., Pidcock, C. J., and Fraser, M. E. (2006). The cerevoice blizzard entry 2006: A prototype database unit selection engine. In *Blizzard Challenge Workshop*, Pittsburgh. 55
- Bangalore, S., Di Fabbrizio, G., and Stent, A. (2006a). Towards learning to converse: Structuring task-oriented human-human dialogs. *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, 1:I-I. 23
- Bangalore, S., Fabbrizio, G. D., and Stent, A. (2006b). Learning the structure of task-driven human-human dialogs. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 201–208, Morristown, NJ, USA. Association for Computational Linguistics. 18, 23
- Barnett, J., Akolkar, R., Auburn, R. J., Bodell, M., Burnett, D. C., Carter, J., McGlashan, S., Helbing, T. L. M., Hosn, R., Raman, T., and Reifenrath, K. (2008). State Chart XML (SCXML) : State machine notation for control abstraction. <http://www.w3.org/TR/scxml/>. 37
- Barzilay, R., Elhadad, N., and McKeown, K. (2002). Inferring strategies for sentence ordering in multi-document summarization. *JAIR*, 17:35–55. 93
- Barzilay, R. and Lapata, M. (2005). Modeling local coherence: An entity-based approach. In *Proc. ACL-05*. 24, 25, 93
- Barzilay, R. and Lee, L. (2004). Catching the drift: probabilistic content models, with applications to generation and summarization. In *In HLT-NAACL 2004: Proceedings of the Main Conference*, pages 113–120. 25
- Berger, A. L., Pietra, V. J. D., and Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71. 70
- Bickmore, T. W., Pfeifer, L. M., and Jack, B. W. (2009). Taking the time to care: empowering low health literacy hospital patients with virtual nurse agents. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, pages 1265–1274, New York, NY, USA. ACM. 2
- Bohus, D. and Rudnicky, A. (2003). Ravenclaw: Dialog management using hierarchical task decomposition and an expectation agenda. In *proceedings of Eurospeech-2003*, Geneva, Switzerland. 5, 16, 21
- Bratt, H., Dowding, J., and Hunicke-Smith, K. (1995). The sri telephone-based atis system. In *Proceedings of the Spoken Language Systems Technology Workshop*. vi, 94, 95
- Brown, G. and Yule, G. (1983). *Discourse Analysis*. Cambridge University Press. 105
- Brown, P. E., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19:263–311. 25
- Bunt, H. (2006). Dimensions in dialogue act annotation. In *Proceedings of of Fifth International Conference on Language Resources and Evaluation (LREC)*. 13
- Bunt, H., Alexandersson, J., Choe, J.-W., Fang, A. C., Hasida, K., Petukhova, V., Popescu-Belis, A., and Traum, D. (2012). Iso 24617-2: A semantically-based standard for dialogue annotation. In Chair), N. C. C., Choukri, K., Declerck, T., Doan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA). 13

- Callison-Burch, C., Osborne, M., and Koehn, P. (2006). Re-evaluating the role of bleu in machine translation research. In *proceedings of EACL-2006*. 102
- Carletta, J., Isard, S., Doherty-Sneddon, G., Isard, A., Kowtko, J. C., and Anderson, A. H. (1997). The reliability of a dialogue structure coding scheme. *Computational Linguistics*, 23(1):13–31. 23
- Chotimongkol, A. and Rudnicky, A. (2002). Automatic concept identification in goal-oriented conversations. In *Proceedings of ICSLP 2002*, pages 1153–1156, Denver, Colorado. 23
- Chotimongkol, A. and Rudnicky, A. (2008). Acquiring domain-specific dialog information from task-oriented human-human interaction through an unsupervised learning. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 955–964, Honolulu, Hawaii. Association for Computational Linguistics. 18, 23
- Chu-carroll, J. (1998). A statistical model for discourse act recognition in dialogue interactions. In *Applying Machine Learning to Discourse Processing. Papers from the 1998 AAAI Spring Symposium. Technical Report SS-98-01*, pages 12–17. AAAI Press. 23
- Chu-Carroll, J. and Carpenter, B. (1999). Vector-based natural language call routing. *Journal of Computational Linguistics*, 25(30):361–388. 7
- Collins, M. (2002). Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics. 63, 71
- Collins, M. (2004). *Parameter estimation for statistical parsing models: theory and practice of distribution-free methods*, pages 19–55. Kluwer Academic Publishers, Norwell, MA, USA. 71
- Core, M. G. and Allen, J. F. (1997). Coding dialogs with the damsl annotation scheme. In *In Proceedings of AAAI97 Fall Symposium on Communicative Action in Humans and Machines*, AAAI. 13, 34
- Dahlbäck, N., Jönsson, A., and Ahrenberg, L. (1993). Wizard of oz studies: why and how. In *IUI '93: Proceedings of the 1st international conference on Intelligent user interfaces*, pages 193–200, New York, NY, USA. ACM. 9
- DeVault, D., Leuski, A., and Sagae, K. (2011a). An evaluation of alternative strategies for implementing dialogue policies using statistical classification and rules. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP)*. 70
- DeVault, D., Leuski, A., and Sagae, K. (2011b). Toward learning and evaluation of dialogue policies with text examples. In *Proceedings of the SIGDIAL 2011 Conference*, pages 39–48, Portland, Oregon. Association for Computational Linguistics. 19, 49, 79, 89, 90
- Eckert, W., Levin, E., and Pieraccini, R. (1997). User modeling for spoken dialogue system evaluation. *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, pages 80–87. 25, 87
- Edlund, J., Gustafson, J., Heldner, M., and Hjalmarsson, A. (2008). Towards human-like spoken dialogue systems. *Speech Commun.*, 50(8-9):630–645. 3

- Elsner, M., Austerweil, J., and Charniak, E. (2007). A unified local and global model for discourse coherence. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 436–443, Rochester, New York. Association for Computational Linguistics.
- Forbes-Riley, K. and Litman, D. J. (2006). Modelling user satisfaction and student learning in a spoken dialogue tutoring system with generic, tutoring, and user affect parameters. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, HLT-NAACL '06*, pages 264–271, Stroudsburg, PA, USA. Association for Computational Linguistics. 10
- Freund, Y. and Schapire, R. E. (1999). Large margin classification using the perceptron algorithm. *Mach. Learn.*, 37:277–296. 71
- Gandhe, S., DeVault, D., Roque, A., Martinovski, B., Artstein, R., Leuski, A., Gerten, J., and Traum, D. (2008). From domain specification to virtual humans: An integrated approach to authoring tactical questioning characters. In *proceedings of Interspeech 2008*. 5, 16, 30, 48
- Gandhe, S., Gordon, A., and Traum, D. (2006). Improving question-answering with linking dialogues. In *International Conference on Intelligent User Interfaces (IUI)*. 26
- Gandhe, S., Rushforth, M., Aggarwal, P., and Traum, D. R. (2011). Evaluation of an integrated authoring tool for building advanced question-answering characters. In *INTERSPEECH*, pages 1289–1292. ISCA. 17, 42
- Gandhe, S. and Traum, D. (2007a). Creating spoken dialogue characters from corpora without annotations. In *Proceedings of Interspeech-07*. 18, 76
- Gandhe, S. and Traum, D. (2007b). First steps towards dialogue modeling from an un-annotated human-human corpus. In *5th Workshop on knowledge and reasoning in practical dialogue systems*, Hyderabad, India. 18, 75
- Gandhe, S. and Traum, D. (2008). Evaluation understudy for dialogue coherence models. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, pages 172–181, Columbus, Ohio. Association for Computational Linguistics. 19
- Gandhe, S. and Traum, D. (2010). I’ve said it before, and i’ll say it again: an empirical investigation of the upper bound of the selection approach to dialogue. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGDIAL '10*, pages 245–248, Stroudsburg, PA, USA. Association for Computational Linguistics. 17, 60
- Gandhe, S. and Traum, D. (2013). Surface text based dialogue models for virtual humans. In *Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Metz, France. Association for Computational Linguistics. 18
- Georgila, K., Henderson, J., and Lemon, O. (2006). User simulation for spoken dialogue systems: Learning and evaluation. In *proceedings of Interspeech*. 26
- Goddeau, D., Meng, H., Polifroni, J., Seneff, S., and Busayapongchai, S. (1996). A form-based dialogue manager for spoken language applications. *Proceedings of Fourth International Conference on Spoken Language Processing, ICSLP 96.*, 2:701–704. 6, 21, 22

- Godfrey, J. J. and Holliman, E. (1993). Switchboard-1 transcripts. Linguistic Data Consortium, Philadelphia. 25
- Godfrey, J. J., Holliman, E. C., and McDaniel, J. (1992). Switchboard: Telephone speech corpus for research and development. In *Proc. of ICASSP-92*, pages 517–520. 59
- Grosz, B. J., Weinstein, S., and Joshi, A. K. (1995). Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21:203–225. 24
- Gustafson, J., Bell, L., Boye, J., Lindström, A., and Wirén, M. (2004). The nice fairy-tale game system. In Strube, M. and Sidner, C., editors, *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, pages 23–26, Cambridge, Massachusetts, USA. Association for Computational Linguistics. 3, 4
- Halliday, M. A. K. and Hasan, R. (1976). *Cohesion in English*. London: Longman. 105
- Heeman, P. (2007). Combining reinformation learning with information-state update rules. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 268–275, Rochester, New York. Association for Computational Linguistics. 8
- Heeman, P. A. and Allen, J. (1994). The TRAINS 93 dialogues. TRAINS Technical Note 94-2, Department of Computer Science, University of Rochester. 59
- Henderson, J., Lemon, O., and Georgila, K. (2005). Hybrid reinforcement/supervised learning for dialogue policies from communicator data. In *proceedings of IJCAI workshop*. 26
- Hone, K. S. and Graham, R. (2000). Towards a tool for the subjective assessment of speech system interfaces (sassi). *Natural Language Engineering: Special Issue on Best Practice in Spoken Dialogue Systems*. 25
- Hunt, A. and Black, A. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 1, pages 373 –376 vol. 1. 55
- Hutchens, J. (1998). how megahal works? <http://megahal.alioth.debian.org/How.html>.
- Ibrahim, A. and Johansson, P. (2002). Multimodal dialogue systems for interactive tv applications. In *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*, pages 117 – 122. 3
- Jan, D. and Traum, D. (2005). Dialog simulation for background characters. In *proceedings of 5th International Working Conference on Intelligent Virtual Agents*. 87
- Jekat, S., Klein, A., Maier, E., Maleck, I., Mast, M., and Quantz, J. J. (1995). Dialogue acts in verbmobil. Technical Report Verbmobil Report 65, Universitat Hamburg, DFKI Saarbrücken and Universitat Erlangen, TU Berlin. 13
- Jelinek, F. and Mercer, R. L. (1980). Interpolated estimation of markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, Amsterdam, The Netherlands: North-Holland. 69

- Jonsson, A. and Dahlback, N. (2000). Distilling dialogues - a method using natural dialogue corpora for dialogue systems development. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, pages 44–51, Seattle, Washington, USA. Association for Computational Linguistics. 86
- Jordan, P., Hall, B., Ringenber, M., Cue, Y., and Rose, C. (2007). Tools for authoring a dialogue agent that participates in learning studies. In *proceedings of AIED 2007*, pages 43–50. 22
- Jurafsky, D. and Martin, J. H. (2000). *SPEECH and LANGUAGE PROCESSING: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall. 25
- Jurafsky, D., Shriberg, E., and Biasca, D. (1997). Switchboard swbd-damsl shallow-discourse-function annotation coders manual. Technical report, University of Colorado Institute of Cognitive Science, Boulder, CO. 13
- Kay, M., Norvig, P., and Gawron, M. (1994). *Verbmobil: A Translation System for Face-to-Face Dialog*. University of Chicago Press, Chicago, IL, USA. 1
- Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, 30:81–93. 94
- Kenny, P., Parsons, T., Gratch, J., Leuski, A., and Rizzo, A. (2007). Virtual patients for clinical therapist skills training. In Pelachaud, C., Martin, J.-C., Andr, E., Chollet, G., Karpouzis, K., and Pel, D., editors, *Intelligent Virtual Agents*, volume 4722 of *Lecture Notes in Computer Science*, pages 197–210. Springer Berlin / Heidelberg. 58
- Khooshabeh, P., McCall, C., Gandhe, S., Gratch, J., and Blascovich, J. (2011). Does it matter if a computer jokes? In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems, CHI EA '11*. 17, 28, 44, 49
- Kita, K., Fukui, Y., Nagata, M., and Morimoto, T. (1996). Automatic acquisition of probabilistic dialogue models. *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, 1:196–199 vol.1. 22
- Krippendorff, K. (2004). *Content Analysis, An Introduction to Its Methodology 2nd Edition*. Sage Publications. 43, 83, 99
- Kronlid, F. and Lager, T. (2007). Implementing the information-state update approach to dialogue management in a slightly extended SCXML. In *Proceedings of the SEMDIAL*. 37
- Lane, H. C., Schneider, M., Michael, S., Albrechtsen, J., and Meissner, C. (2010). Virtual humans with secrets: Learning to detect verbal cues to deception. In Aleven, V., Kay, J., and Mostow, J., editors, *Intelligent Tutoring Systems*, volume 6095 of *Lecture Notes in Computer Science*, pages 144–154. Springer Berlin / Heidelberg. 28, 44, 45, 49
- Lapata, M. (2003). Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan. 19, 93
- Lapata, M. (2006). Automatic evaluation of information ordering. *Computational Linguistics*, 32(4):471–484. 93, 94, 95, 97, 105
- Lapata, M. and Barzilay, R. (2005). Automatic evaluation of text coherence: Models and representations. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 1085–1090, Edinburgh.

- Larsson, S. (2002). Issues under negotiation. In *Proceedings of the 3rd SIGdial workshop on Discourse and dialogue*, pages 103–112, Morristown, NJ, USA. Association for Computational Linguistics. 14
- Larsson, S. (2005). Dialogue systems: Simulations or interfaces? In Gardent and Gaiffe, editors, *Proceedings of the ninth workshop on the semantics and pragmatics of dialogue*. 3
- Larsson, S., Berman, A., Hallenborg, J., and Hjelm, D. (2004). Trindikit 3.1 manual. Technical report, Department of Linguistics, Goteborg University. 16, 22
- Larsson, S. and Traum, D. R. (2000). Information state and dialogue management in the trindi dialogue move engine toolkit. *Nat. Lang. Eng.*, 6(3-4):323–340. 6, 22, 37
- Lavie, A. and Denkowski, M. J. (2009). The meteor metric for automatic evaluation of machine translation. *Machine Translation*, 23:105–115. 59
- Lavrenko, V. (2004). *A Generative Theory of Relevance*. PhD thesis, University of Massachusetts at Amherst. 69
- Lavrenko, V., Choquette, M., and Croft, W. B. (2002). Cross-lingual relevance models. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, pages 175–182, New York, NY, USA. ACM. 25, 63, 68
- Lee, C., Jung, S., Eun, J., Jeong, M., and Lee, G. G. (2006). A situation-based dialogue management using dialogue examples. *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, 1:I–I. 18, 24
- Lee, C., Jung, S., and Lee, G. G. (2008). Robust dialog management with n-best hypotheses using dialog examples and agenda. In *Proceedings of ACL-08: HLT*, pages 630–637, Columbus, Ohio. Association for Computational Linguistics. 24
- Lee, J. and Marsella, S. (2006). Nonverbal behavior generator for embodied conversational agents. In *Proceeding of Intelligent Virtual Agents*, pages 243–255. 55
- Leuski, A., Patel, R., Traum, D., and Kennedy, B. (2006). Building effective question answering characters. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pages 18–27, Sydney, Australia. Association for Computational Linguistics. 4, 5, 7, 16, 22, 25, 28, 44, 58, 59, 63, 68, 86
- Leuski, A. and Traum, D. (2008). A statistical approach for text processing in virtual humans. In *Proceedings of 26th Army Science Conference*. 35
- Leuski, A. and Traum, D. (2010). Practical language processing for virtual humans. In *Proceedings of the Twenty-Second Annual Conference on Innovative Applications of Artificial Intelligence (IAAI-10)*, pages 1740–1747. 69
- Leuski, A. and Traum, D. (2011). Npceditor: Creating virtual human dialogue using information retrieval techniques. *AI Magazine*, 32(2):42–56. 22, 70
- Levin, E., Pieraccini, R., and Eckert, W. (1997). Learning dialogue strategies within the markov decision process framework. *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, pages 72–79. 25, 62, 87
- Levin, E., Pieraccini, R., and Eckert, W. (2000). A stochastic model of human-machine interaction for learning dialog strategies. *Speech and Audio Processing, IEEE Transactions on*, 8(1):11–23. 8, 23, 24, 105

- Lewin, I. (2000). A formal model of conversational game theory. In *Fourth Workshop on the Semantics and Pragmatics of Dialogue: Gotalog 2000*, pages 115–122. 19, 37
- Lin, C.-Y. and Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 71–78, Morristown, NJ, USA. Association for Computational Linguistics. 89
- Litman, D. J. and Silliman, S. (2004). ITSPOKE: An Intelligent Tutoring Spoken Dialogue System. In Susan Dumais, D. M. and Roukos, S., editors, *HLT-NAACL 2004: Demonstration Papers*, pages 5–8, Boston, Massachusetts, USA. Association for Computational Linguistics. 1
- Margaretha, E. and DeVault, D. (2011). An approach to the automated evaluation of pipeline architectures in natural language dialogue systems. In *Proceedings of the SIGDIAL 2011 Conference*, pages 279–285, Portland, Oregon. Association for Computational Linguistics. 49
- Marinelli, D. and Stevens, S. (1998). Synthetic interviews: the art of creating a ‘dyad’ between humans and machine-based characters. In *Interactive Voice Technology for Telecommunications Applications, 1998. IVTTA '98. Proceedings. 1998 IEEE 4th Workshop*, pages 43–48. 7
- McGlashan, S., Burnett, D. C., Carter, J., Danielsen, P., Ferrans, J., Hunt, A., Lucas, B., Porter, B., Rehor, K., and Tryphonas, S. (2004). Voice extensible markup language (voicexml) version 2.0. <http://www.w3.org/TR/voicexml20/>. 21
- Minami, Y., Mori, A., Meguro, T., Higashinaka, R., Dohsaka, K., and Maeda, E. (2011). Dialogue control by pomdp using dialogue data statistics. In Minker, W., Lee, G. G., Nakamura, S., and Mariani, J., editors, *Spoken Dialogue Systems Technology and Design*, pages 163–186. Springer New York. 62
- Mitchell, M. and Sproat, R. (2012). Discourse-based modeling for aac. In *Proceedings of the Third Workshop on Speech and Language Processing for Assistive Technologies*, pages 9–18, Montréal, Canada. Association for Computational Linguistics. 88
- MITRE, C. (2005). Midiki: Mitre dialogue kit user’s manual. Technical report, The MITRE Corporation. 16, 22
- Narayanan, S., Ananthakrishnan, S., Belvin, R., Ettelaie, E., Gandhe, S., Ganjavi, S., Georgiou, P. G., Hein, C. M., Kadambe, S., Knight, K., Marcu, D., Neely, H. E., Srinivasamurthy, N., Traum, D., and Wang, D. (2004). The transonics spoken dialogue translator: An aid for english-persian doctor-patient interviews. In *working notes of the AAAI Fall symposium on Dialogue Systems for Health Communication*, pages 97–103. 1
- Och, F. J. (2003). Minimum error rate training for statistical machine translation. In *In ACL 2003: Proc. of the 41st Annual Meeting of the Association for Computational Linguistics*. 105
- Paek, T. and Pieraccini, R. (2008). Automating spoken dialogue management design using machine learning: An industry perspective. *Speech Communication*, 50(8-9):716 – 729. Evaluating new methods and models for advanced speech-based interactive systems. 23
- Papineni, K. A., Roukos, S., Ward, T., and Zhu, W.-J. (2001). Bleu: a method for automatic evaluation of machine translation. In *Technical Report RC22176 (W0109-022), IBM Research Division*. 89

- Poesio, M. and Traum, D. (1998). Towards an axiomatization of dialogue acts. In *Proceedings of the Twente Workshop on the Formal Semantics and Pragmatics of Dialogues (13th Twente Workshop on Language Technology)*, pages 207–222. 15
- Prasad, R. and Walker, M. (2004). 2000 communicator dialogue act tagged. Linguistic Data Consortium, Philadelphia. 2
- Purandare, A. and Litman, D. (2008). Content-learning correlations in spoken tutoring dialogs at word, turn and discourse levels. In *Proceedings 21st International FLAIRS Conference*. 25, 94
- Raux, A., Langner, B., Bohus, D., Black, A. W., and Eskenazi, M. (2005). Lets go public! taking a spoken dialog system to the real world. In *Proceedings of Interspeech 2005*. 1, 4, 5
- Reithinger, N. and Klesen, M. (1997). Dialogue act classification using language models. In *In Proceedings of EuroSpeech-97*, pages 2235–2238. 23
- Rieser, V. and Lemon, O. (2006). Using logistic regression to initialise reinforcement-learning-based dialogue systems. *Spoken Language Technology Workshop, 2006. IEEE*, pages 190–193. 24
- Rieser, V. and Lemon, O. (2008). Learning effective multimodal dialogue strategies from Wizard-of-Oz data: Bootstrapping and evaluation. In *Proceedings of ACL-08: HLT*, pages 638–646, Columbus, Ohio. Association for Computational Linguistics. 23
- Ritter, A., Cherry, C., and Dolan, B. (2010). Unsupervised modeling of twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT '10*, pages 172–180, Stroudsburg, PA, USA. Association for Computational Linguistics. 19
- Roark, B., Saraclar, M., and Collins, M. (2007). Discriminative n-gram language modeling. *Comput. Speech Lang.*, 21:373–392. 71
- Roque, A. (2009). *Dialogue management in spoken dialogue systems with degrees of grounding*. PhD thesis, University of Southern California, Los Angeles, CA, USA. AAI3355296. 34
- Roque, A., Ai, H., and Traum, D. (2006a). Evaluation of an information state-based dialogue manager. In *Brandial 2006: The 10th Workshop on the Semantics and Pragmatics of Dialogue*. 26
- Roque, A., Georgila, K., Artstein, R., Sagae, K., and Traum, D. R. (2010). Natural language processing for joint fire observer training. In *Proceedings of the 27th Army Science Conference*. 59
- Roque, A., Leuski, A., Sridhar, V. K. R., Robinson, S., Vaswani, A., Narayanan, S., and Traum, D. R. (2006b). Radiobot-cff: a spoken dialogue system for military training. In *INTERSPEECH*, Pittsburgh, PA, USA. 59
- Roque, A. and Traum, D. (2007). A model of compliance and emotion for potentially adversarial dialogue agents. In *The 8th SIGdial Workshop on Discourse and Dialogue*. 29, 40, 48
- Roque, A. and Traum, D. (2009). Improving a virtual human using a model of degrees of grounding. In *Proceedings of IJCAI-09*. 39
- Rushforth, M., Gandhe, S., Artstein, R., Roque, A., Ali, S., Whitman, N., and Traum, D. R. (2009). Varying personality in spoken dialogue with a virtual human. In Ruttkay, Z., Kipp, M., Nijholt, A., and Vilhjálmsson, H. H., editors, *Intelligent Virtual Agents: 9th International Conference, IVA 2009, Amsterdam, The Netherlands, September 14–16, 2009 Proceedings*, volume 5733 of *Lecture Notes in Artificial Intelligence*, page 541–542, Heidelberg. Springer, Springer. 40

- Salton, G. and McGill, M. (1983). *Introduction to modern information retrieval*. McGraw-Hill computer science series. McGraw-Hill. 64
- Samuel, K., Carberry, S., and Vijay-Shanker, K. (1998). Dialogue act tagging with transformation-based learning. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, pages 1150–1156, Montreal, Quebec, Canada. Association for Computational Linguistics. 23
- Schatzmann, J., Thomson, B., Weilhammer, K., Ye, H., and Young, S. (2007). Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *proceedings of HLT/NAACL*. 24, 26
- Searle, J. R. (1969). *Speech Acts*. Cambridge University Press. 5
- Sellberg, L. and Jönsson, A. (2008). Using random indexing to improve singular value decomposition for latent semantic analysis. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Morocco. 58
- Seneff, E., Hirschman, L., and Zue, V. (1991). Interactive problem solving and dialogue in the atis domain. In *Proceedings of the Pacific Grove Workshop*, pages 354–359. 3
- Siu, K.-C. and Meng, H. M. (1999). Semi-automatic acquisition of domain-specific semantic structures. In *EUROSPEECH'99*, pages 2039–2042. 23
- Soricut, R. and Marcu, D. (2006). Discourse generation using utility-trained coherence models. In *Proc. ACL-06*. 25, 93
- Stalnaker, R. (1973). Presuppositions. *Journal of Philosophical Logic*, 2:447–457. 66
- Stent, A. J. (2000). The monroe corpus. Technical Report 728, Computer Science Dept. University of Rochester. 59
- Stolcke, A., Coccaro, N., Bates, R., Taylor, P., Ess-Dykema, C. V., Ries, K., Shriberg, E., Jurafsky, D., Martin, R., and Meteer, M. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373. 18, 23
- Sutton, S., Cole, R., Villiers, J. D., Schalkwyk, J., Vermeulen, P., Macon, M., Yan, Y., Rundle, B., Shobaki, K., Hosom, P., Kain, A., Wouters, J., Massaro, D., and Cohen, M. (1998). Universal speech tools: The CSLU toolkit. In *In Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 3221–3224. 16, 21
- Swartout, W., Traum, D., Artstein, R., Noren, D., Debevec, P., Bronnenkant, K., Williams, J., Leuski, A., Narayanan, S., Piepol, D., Lane, C., Morie, J., Aggarwal, P., Liewer, M., Chiang, J.-Y., Gerten, J., Chu, S., and White, K. (2010). Ada and grace: toward realistic and engaging virtual museum guides. In *Proceedings of the 10th international conference on Intelligent virtual agents, IVA'10*, pages 286–300, Berlin, Heidelberg. Springer-Verlag. 2
- Traum, D. (1999). Speech acts for dialogue agents. In Wooldridge, M. and Rao, A., editors, *Foundations of Rational Agency*, pages 169–201. Kluwer. 6
- Traum, D. (2000). 20 Questions on Dialogue Act Taxonomies. *J Semantics*, 17(1):7–30. 13
- Traum, D. and Larsson, S. (2003). The information state approach to dialogue management. In van Kuppevelt, J. and Smith, R., editors, *Current and New Directions in Discourse and Dialogue*. Kluwer. 9, 19, 22, 30, 38

- Traum, D., Leuksi, A., Roque, A., Gandhe, S., DeVault, D., Gerten, J., Robinson, S., and Martinovski, B. (2008a). Natural language dialogue architectures for tactical questioning characters. In *Proceedings of 26th Army Science Conference*. 2, 28
- Traum, D., Marsella, S. C., Gratch, J., Lee, J., and Hartholt, A. (2008b). Multi-party, multi-issue, multi-strategy negotiation for multi-modal virtual agents. In *Proceedings of the 8th international conference on Intelligent Virtual Agents, IVA '08*, pages 117–130, Berlin, Heidelberg. Springer-Verlag. 7, 59
- Traum, D., Roque, A., Leuski, A., Georgiou, P., Gerten, J., Martinovski, B., Narayanan, S., Robinson, S., and Vaswani, A. (2007). Hassan: A virtual human for tactical questioning. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, pages 71–74. 30
- Traum, D., Swartout, W., Gratch, J., and Marsella, S. (2005). Virtual humans for non-team interaction training. In *AAMAS-05 Workshop on Creating Bonds with Humanoids*. 2, 4, 5, 10, 14, 26, 54, 77, 95
- Traum, D., Swartout, W., Gratch, J., and Marsella, S. (2008c). *A Virtual Human Dialogue Model for Non-Team Interaction*, volume 39 of *Text, Speech and Language Technology*, pages 45–67. Springer Netherlands. 1, 2, 87
- Traum, D. R. and Allen, J. F. (1994). Discourse obligations in dialogue processing. In *proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL-94)*, pages 1–8. 37, 101
- Traum, D. R., Robinson, S., and Stephan, J. (2004). Evaluation of multi-party virtual reality dialogue interaction. In *In Proceedings of Fourth International Conference on Language Resources and Evaluation (LREC)*, pages 1699–1702. 26
- Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59:433–460. 12
- Walker, M., Aberdeen, J., Boland, J., Bratt, E., Garofolo, J., Hirschman, L., Le, A., Lee, S., Narayanan, S., Papineni, K., Pellom, B., Polifroni, J., Potamianos, A., Prabhu, P., Rudnick, A., Sanders, G., Sen-eff, S., Stallard, D., , and Whittaker, S. (2001). Darpa communicator dialog travel planning systems: The june 2000 data collection. In *Proceedings of EUROSPEECH 2001*. 1, 3, 4, 5
- Walker, M., Kamm, C., and Litman, D. (2000). Towards developing general models of usability with paradise. *Natural Language Engineering: Special Issue on Best Practice in Spoken Dialogue Systems*. 10, 24, 25, 91
- Walker, M. and Passonneau, R. (2001). Date: a dialogue act tagging scheme for evaluation of spoken dialogue systems. In *Proceedings of the first international conference on Human language technology research, HLT '01*, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics. 7
- Wallace, R. (2003a). *AIML Overview*. ALICE A. I. Foundation. 22
- Wallace, R. (2003b). *Be Your Own Botmaster, 2nd Edition*. ALICE A. I. Foundation. 3, 4, 7, 26
- Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45. 1, 3, 4, 7, 26
- Weng, F., Varges, S., Raghunathan, B., Ratiu, F., Pon-Barry, H., Lathrop, B., Zhang, Q., Bratt, H., Scheideck, T., Xu, K., Purver, M., Mishra, R., Lien, A., Raya, M., Peters, S., Meng, Y., Russell, J., Cavedon, L., Shriberg, E., Schmidt, H., and Prieto, R. (2006). CHAT: a conversational helper for automotive tasks. In *INTERSPEECH'06*. 3

- Williams, J. D. (2007). A method for evaluating and comparing user simulations: The cramer-von mises divergence. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Kyoto, Japan. 26
- Williams, J. D. and Young, S. (2003). Using wizard-of-oz simulations to bootstrap reinforcement - learning based dialog management systems. In Kurematsu, A., Rudnicky, A., and Tutiya, S., editors, *Proceedings of the Fourth SIGdial Workshop on Discourse and Dialogue*, pages 135–139. 24
- Williams, J. D. and Young, S. (2007). Partially observable markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21:393–422. 8, 24, 25, 62, 105
- Woszczyna, M. and Waibel, A. (1994). Inferring linguistic structure in spoken language. In *ICSLP-1994*, pages 847–850. 22
- Xu, W. and Rudnicky, A. I. (2000). Task-based dialog management using an agenda. In *ANLP/NAACL 2000 Workshop on Conversational systems*, pages 42–47, Morristown, NJ, USA. Association for Computational Linguistics. 6, 22
- Yao, X., Bhutada, P., Georgila, K., Sagae, K., Artstein, R., and Traum, D. R. (2010). Practical evaluation of speech recognizers for virtual human dialogue systems. In *LREC-2010*, Valetta, Malta. 49
- Zue, V., Sene, S., Glass, J., Polifroni, J., Pao, C., Hazen, T. J., and Hetherington, L. (2000). JUPITER: a telephone-based conversational interface for weather information. *IEEE Transactions on Speech and Audio Processing*, 8(1):85 –96. 5
- Zukerman, I. and Marom, Y. (2006). A corpus-based approach to help-desk response generation. In *Computational Intelligence for Modelling, Control and Automation (CIMCA 2006)*, *International Conference on Intelligent Agents, Web Technologies and Internet Commerce (IAWTIC 2006)*. 58