NATURAL LANGUAGE DIALOGUE ARCHITECTURES FOR TACTICAL QUESTIONING CHARACTERS

David Traum, Anton Leuski, Antonio Roque, Sudeep Gandhe,
David DeVault, Jillian Gerten, Susan Robinson, Bilyana Martinovski
Institute for Creative Technologies
University of Southern California
Marina del Rey, CA, 90292, USA
{traum}@ict.usc.edu

ABSTRACT

In this paper we contrast three architectures for natural language questioning characters. We contrast the relative costs and benefits of each approach in building characters for tactical questioning. The first architecture works purely at the textual level, using cross-language information retrieval techniques to learn the best output for any input from a training set of linked questions and answers. The second architecture adds a global emotional model and computes a compliance model, which can result in different outputs for different levels, given the same inputs. The third architecture works at a semantic level and allows authoring of different policies for response for different kinds of information. We describe these architectures and their strengths and weaknesses with respect to expressive capacity, performance, and authoring demands.

1. INTRODUCTION

A natural language dialogue system receives natural language input from a user and communicates in natural language to the user. There are many different types of dialogue systems which vary along a number of dimensions. Some of these dimensions include:

- The modalities for communication
- The task
- The depth of representation
- The use of context
- the architecture of the system
- the algorithms used for processing
- the means for giving the system specific knowledge of the situation it is talking about

We discuss each of these in turn. There are many possible modalities that can be used for communicating. On the input from the user these could include typed text, handwritten text, speech, GUI manipulations, or physical gestures. On the output side the system could speak, display text, display graphical elements in a GUI, or use the gestures of a robot or virtual character. Any combination of one or more input and output modalities is possible.

One of the most important issues is what the conversation is about. Is this just a casual conversation to pass the time, or is there some kind of task that the conversation is oriented towards? Is the task going on at the same time, before, or after the conversation? How complex is the task? How closely related is the conversation to the task? Is the task a physical one, such as assembling an artifact, or an informational one, such as database look-up? Can the task be done by both participants equally or is there asymmetric access to the actions of the task? The answers to these questions will have important implications on the requirements of the system. For instance, will it just have to answer questions, or will it have to ask questions, make suggestions, or negotiate?

There are different levels at which to focus the processing of inputs and outputs. The system could reason just about the words that are communicated by the user or deeper structures and meanings. If the system considers the words, it may focus on just important keywords, or a specific syntactic configuration or statistical distribution of words. Realizing that many combinations of word sequences can mean the same thing and small differences in the set or order of the words may mean different things, it may be beneficial to do reasoning at a deeper level, where many small differences in wording can be collapsed to the same meaning.

Context can be very important for processing language, particularly in dialogue. There are a number of different types of context that may be available for constraining or altering the meanings of natural language expressions. First, there is the context of what has been said before. Virtually all dialogue systems use at least the context of the user's previous utterance in deciding what to say. However it may often be important to look further back. Also important is the context of physical or virtual space that the conversants are embedded in. Are there things that they can both hear, see, smell, or feel? There is also the intensional space - what kind of beliefs, plans, thoughts, feelings, intentions, or inference patterns are common to both and can be referred to implicitly in conversation? As well as the shared context there is also individual context, such as the internal state of the participants. Often what someone says has more to do with this internal state than what another has just said.

There are many different ways to architect a system. Some key design questions include: how many functions does the system perform? Are the processing of these functions unified into a single software module or distributed into many modules? What are the specific modules, and how do they pass information? Are they serialized into a pipeline, or are they working in parallel or communicating information back and forth to make decisions?

At least at the present time, there is no single best architecture for all dialogue systems. For different combinations of task and available/necessary communication modalities, there can be different optimal architectures and amounts of representation and context usage. Moreover, for different architectures and these factors there can be different algorithms that are best. There are also different factors for deciding on optimality, including at least:

- System coverage
- system accuracy
- System run-time speed
- User satisfaction
- System Design time
- Training needed to use
- training needed to design
- any combination of the above.

In this paper we examine one small aspect of this general problem, by holding the domain and modalities constant, and examining three different architectures which support different degrees of use of context and depth of representation, and examining the resulting systems with respect to the evaluation factors above, particularly, system design time and ease, coverage and accuracy.

In the next section we introduce the Tactical Questioning genre and our testing domain, the virtual character Hassan, who has been implemented in all three architectures. In section 3, we describe the first architecture, based directly on [Leuski *et al.*, 2006c]. In section 4, we describe the second architecture, presented more fully in [Traum *et al.*, 2007, Roque and Traum, 2007]. In section 5, we describe the third architecture, described more fully in [Gandhe *et al.*, 2008]. In section 6, we compare and contrast these architectures and the resulting systems. Finally, we conclude in Section 7.

2. TACTICAL QUESTIONING AND HASSAN

Tactical Questioning dialogues are those in which small-unit military personnel, usually on patrol, hold conversations with individuals to produce information of military value [Army, 2006]. We are specifically interested in this domain when applied to civilians, when the process becomes more conversational and additional goals involve building rapport with the population and gathering general information about the area of operations. Hassan (see Figure 1) is a virtual human designed to act as a roleplayer and allow trainees to practice tactical questioning and get feedback from experienced instructors on their performance on several learning goals.

Figure 1: Hassan

The scenario for Hassan takes place in contemporary Iraq. In a fictional storyline, the US authorities have built a marketplace as part of the reconstruction effort, but the local population continues to use the old, broken-down marketplace instead. It is the goal of the trainee to discover why. To do this, the trainee talks to Hassan, a local politician. If the trainee convinces Hassan to help him, the trainee will confirm that a tax has been levied on the new marketplace, and that the tax has been placed by Hassan's employer; if exceptionally successful, the trainee may even learn where that employer lives.

The domain of training tactical questioning has several important implications for a dialogue system. In our case, there is a multi-modal communication with an embodied agent, such as Hassan, shown in Figure 1. This requires speech and non-verbal communication. The task is one of questions and answers, which has less reasoning and inference demands than some domains involving detailed planning and negotiation, though still has some demands in this regard, particularly decisions about which questions to answer and what the implications of certain kinds of answers are. On the other hand, the domain is not severely limited in scope as a transaction oriented task would be. The user/trainee could ask Hassan any question at all, and Hassan must try to respond appropriately (even if this is just a complaint that the question is inappropriate). Context can be important, especially a knowledge of what has previously been established in the conversation, but also the evolving internal state of the character, as one of the main goals of questioning is to get the one being questioned to the point where they will reveal important information; a good training application will reward productive lines of inquiry and fail to reward inappropriate or ineffective behavior. In this case we also had a goal of developing technology that educators who are expert at questioning but not necessarily experts at dialogue system technology could use to develop scenarios.

For embodiment of the virtual human and production of gestures and non-verbal behaviors we used the same components for all three versions - we re-used aspects of the Virtual human architecture [Kenny *et al.*, 2007a] including Smartbody [Thiebaux *et al.*, 2008] and the non-verbal behavior generator [Lee and Marsella, 2006]. We also used the same speech recognizer, the Sonic system[Pellom, 2001], augmented with a domain specific language model [Sethy *et al.*, 2006].

3. ARCHITECTURE 1

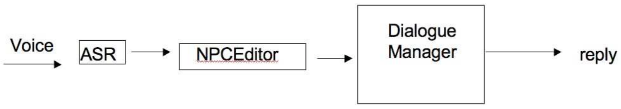

Figure 2: Architecture 1

Given the characteristics of the domain and goals of authorability by non-dialogue system specialists, our starting point for the Hassan character was the question-answering character technology described in [Leuski *et al.*, 2006c]. The language processing architecture is shown in Figure 2. The central component is the NPCEditor, a general tool used for statistical classification of language input and output. It contains an authoring panel in which one can author sample inputs and outputs and link them together (in a many-to-many relationship). It also contains facilities for categorizing questions and answers along several dimensions, methods for specifying the input and output message formats, and protocols to assist dialogue management. The core is a classifier based on cross-language information retrieval that can use the authored information to construct a score and ranking for each possible output given a new input, with reference to language models created from the authored inputs, outputs, and connections. In the simplest configuration, the authored inputs are text questions or initiatives from a user and the outputs are text of the character. This configuration is used in Architecture 1 as well as Sgt Blackwell [Leuski *et al.*, 2006b], Sgt Star [Artstein *et al.*, 2008], and virtual patients [Kenny *et al.*, 2007b, Kenny *et al.*, 2008]. The author will specify the texts of the responses the character will give and a set of questions that might be asked (the initial authoring is usually followed by a data-gathering phase in which the questions of other test users can be added to the question set for broader applicability). The domain model is represented as a property list, for easy swapping in and out of different domains.

There is a very rudimentary dialogue manager in Architecture 1, which merely tries to avoid too much repetition in answers (if possible), chooses a specific prosody for the output speech, and buffers the replies so that the body does not try to blend two speech utterances into one.

Trainee	Hello Hassan
Hassan	Hello
Trainee	You have a very nice house?
Hassan	Thank you, but it was much more magnificent before. We can only repair so much
Trainee	I have some questions about the tax being collected
Hassan	Even if I knew something, would you truly expect me to tell you
Trainee	yes, because I care about the community.
Hassan	You talk to me as if I have something to do about the tax being collected. a soldier came and told us to collect the tax. Now you are angry about it
Trainee	What was the soldiers name
Hassan	you all look the same to me... i cannot remember his name

Figure 3: Dialogue with Architecture 1

This architecture has a number of advantages. First, it is easy to create a simple domain — the author merely has to construct the questions, answers, and links. It can support dialogues such as that in Figure 3. Second, it is very robust to noisy input, which can occur either when a user asks something that is not using the precise words of the authored questions, or when speech recognition makes errors, or both [Leuski *et al.*, 2006a].

Unfortunately there are also some disadvantages of this architecture. First, there is no easy way to vary the answers of the character depending on how the dialogue progresses. Ideally one would have a progression of mood so that the kinds of responses later in the dialogue differ depending on how the earlier parts go, and trainees might find a character more or less compliant depending on how they have conducted the interview. Second, there is not enough memory to maintain consistency of responses. If there are multiple possible answers to a question (say, some truthful and some lies), the NPCEditor does not have any information on which to base a decision, and might come out with a random, unmotivated sequence. Third the authoring burden is fairly high for revising the domain. When a new answer is created it may change the balance of whether previous answers are still reasonable for other questions. In order to do a good job of revising the domain, an author must keep the whole set of questions and answers in mind. This is not a problem for very small domains, but even for a moderate size such as Hassan (600 questions, 140 answers), this becomes problematic, and some domains are much larger.

4. ARCHITECTURE 2

Our second architecture seeks to maintain the simplicity of authoring and direct connection between input question and output answer, yet allow responses to be governed by a multi-component global emotional model and a selected compliance level. This architecture, shown in Figure 4, and described in [Traum *et al.*, 2007] includes a much more extensive dialogue manager, which tracks several emotional variables and decides on a model of compliance for the character, whether it will answer questions honestly, will try to hide information, or will act antagonistically [Roque and Traum, 2007]. Answers for each compliance level are computed by having several different domains, each using an NPCEditor with its own input questions, answers, and links, and the dialogue manager will select the best answer depending on the current compliance level. There are also additional classifiers to identify the topic of discussion, polite utterances, and types of dialogue move (e.g. offer, threat, question), which are used to update the individual emotion components. These classifiers also use the NPCEditor, but in this case, rather than linking questions to answers that the character might give, they link questions to one or another category label.

The dialogue manager computes compliance levels based on configurations of emotional variables such as fear, respect, feels respected, and closeness. Different combinations can lead to compliance and antagonism, modelling success of different kinds of questioning techniques, such as direct approach, incentive and “fear up”. This second architecture can model global dialogue coherence, so that the character answers only when in the right compliance level, and incentives, threats, and politeness can all have positive or negative effects. The authoring is a little more labor intensive - because of the need to author multiple sets of answers and links, however it maintains the same level of simplicity. A more serious problem, though, is that like the first architecture, there is little prospect of local coherence. For instance, if the character solicits an incentive before answering a question, and this incentive is offered, the character does not remember and answer the prior question. If this incentive changes the compliance level, and the questioner asks again the character will now answer, but not otherwise.

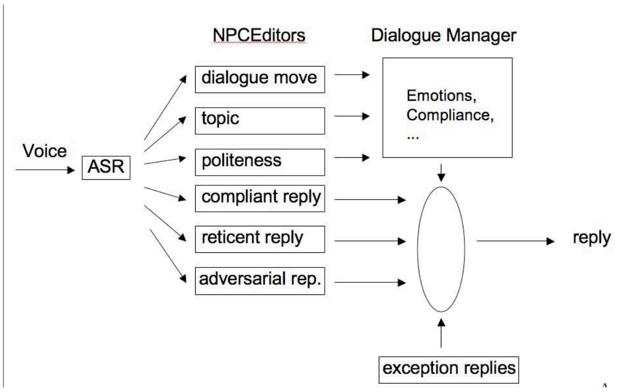

Figure 4: Architecture 2

Figure 5 shows an example dialogue with Hassan using architecture 2. We can infer from Hassan’s answers that he starts out in a reticent mood, since he is reluctant to pass on any information, but not openly hostile. The last utterance seems to indicate more of a shift to an antagonistic level. Other lines of questioning would lead to more compliance. This version of Hassan maintains the same 600 input questions, but now has 122 compliant answers, 62 reticent answers and 90 antagonistic answers (some answers appear in multiple categories).

Trainee	Hello Hassan
Hassan	Hello
Trainee	How are you doing?
Hassan	Well, under the circumstances we are fine
Trainee	I’d like to talk about the marketplace
Hassan	I hope you do not expect me to tell you anything
Trainee	I just want to know why people aren’t using the marketplace
Hassan	I don’t feel like answering that question
Trainee	I think you know something about a tax
Hassan	I am simply doing business. It is rude of you to imply otherwise

Figure 5: Dialogue with Architecture 2

Architecture 2 also has several comparative strengths and weaknesses. It advances the capability of the system, with the compliance model that can handle global coherence, but this comes at a cost of increased authoring burden, since one must author several domains rather than just one

(one for each compliance level). There's also a difficulty in that the feature recognizers are independent of the answer selection mechanism, which can lead to some anomalous results in some cases. Finally, it still does not address the problem of local coherence beyond the question-answer pair and is not able to effectively handle bargaining with incentives and threats about a single piece of information.

5. ARCHITECTURE 3

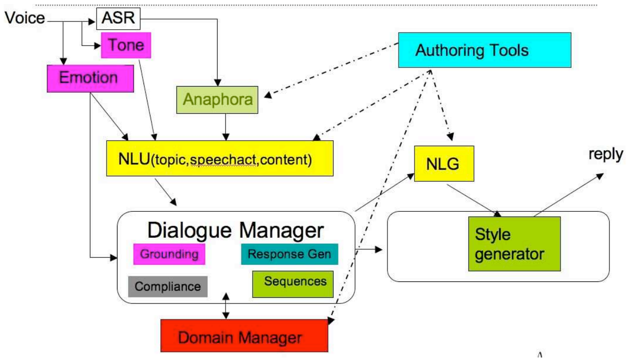

Figure 6: Architecture 3

The third architecture attempts to overcome the deficits of the previous two, while maintaining many of their strengths. This architecture, shown in Figure 6, and described in [Gandhe *et al.*, 2008], allows authoring at the level of basic knowledge that is involved in the domain, while automatically generating dialogue acts that encapsulate changes in the commitments and local context of the dialogue. The author starts the domain creation thinking about the issues that will be talked about, not the precise language for talking about them. Authors then create basic objects with attributes and values, as well as goals, offers, and threats. One can create values that are labelled as either "true" or "false", so that an agent can make a strategic decision about when to lie. From the basic set of objects and values, a set of XML representations of speech acts for both the person and the character are automatically created by the authoring tool, and a user is then able to link these to the input and output sentences.

One can also author both general policies and policies specific to a particular piece of information. For example, one might want a character such as Hassan to always tell the truth about his name, but refuse to answer a sensitive subject such as the name of his superior or who is responsible for the tax, unless some incentives such as safety or monetary reward are given. The authoring tools then export data to several components, Natural language understanding (NLU), Natural language generation (NLG), and the domain and dialogue managers.

The NLU and NLG components are both implemented in the NPCEditor. NLU takes questions or other textual inputs from the user, and produces speech acts. NLG does the reverse: takes speech acts from the dialogue manager and produces text for the character to say. The domain manager represents a set of facts about the domain and communicates with the dialogue manager in interpreting new input, updating the dialogue state and deciding what to say.

A series of finite state networks track the local state for each question or offer and what relevant moves have been made. Decisions of what to say are made at the content level, considering the state of the networks, the compliance level, and other factors such as grounding of information. This allows much more coherent interaction across several exchanges. In the case in which a solicited incentive is provided, a compliant character can respond immediately without needing to be re-prompted. Figure 7 and 8 show examples of short dialogue excerpts that are not possible with the previous architectures. Here the answer is not given directly in response to a question, but only after a relevant incentive has been provided. This version of Hassan can not only answer or reject, but also elicit incentives and then answer only when the incentives are given. The Finite state model allows an arbitrary nesting of other material between the first and third utterances before answering.

Trainee	can you tell us the sheik's identity
Hassan	I would prefer to remain anonymous in these matters.
Trainee	we can let you remain anonymous
Hassan	My Imam's name is Abdullah.

Figure 7: Dialogue with Architecture 3

Trainee	well then why don't you just tell me who is collecting the tax
Hassan	I might tell you what you want if there was something in it for me.
Trainee	now I've already told you that we can protect you or offer you money
Hassan	Indeed, you might say that I collect the taxes.

Figure 8: Another Dialogue with Architecture 3

The dialogue manager in Architecture 3 contains a number of integrated functions. First there is a tracking of the states of all relevant finite state machines, each of which suggests possible next moves. Second, there is the compliance computation, which is an extension of that used in the dialogue manager from architecture 2. In this case, compliance is not used to select a specific answer from one of the NPCEditors, but rather used as an input to other modules in computing transitions from states and the way to respond. There is also a new Grounding module, based on the one in [Roque and Traum, 2008], adapted to the tactical questioning domain. This will decide how the character should indicate its level of understanding through positive or negative feedback, which could be either implicitly or explicitly produced. There is also a response generation component which takes into account the set of candidate outputs from the sequence manager, the grounding policies and the compliance level in formulating a final response.

Architecture 3 also includes some optional performance enhancing components. There is additional information from the speech analysis component including recognizers for speech tone and emotion of the speaker [Busso *et al.*, 2008]. There is also an anaphora component that can track probable meanings for words like "he", "she", "it", "this", "that" and replace them with the likely contents for higher performance of the NLU component. Finally, there is a style generator based on [Devault *et al.*, 2008], that can vary the output based on current aspects of the emotion model, for a richer set of responses that are better tuned to the characters' internal state. Our Hassan character has 89 user speech acts and 85 character speech acts.

Architecture 3 improves on the predecessors in several ways. It can provide much more coherent behavior for small sections of dialogue that all relate to a single topic. There is a much better integrated sets of reasoning and inference in the dialogue manager, since contents and features are recognized together in a single NLU component. The authoring burden is different than in the first two architectures, since one now separately authors the basic knowledge, the policies for deciding on when and how to reveal important information, and the text that will go with both questions and answers. This is arguably a higher requirement on the author, since more knowledge is needed about the knowledge level, as well as the text level. On the other hand, we feel that the overall burden on the author is lower, since one need not make as many decisions about specific questions and answers up-front. Our approach is to provide two levels of capability. First, "power-user" tools that allow someone with very precise demands to individually author things like policies and emotional models. And second, a normal user capability with a set of default policies and behaviors to choose from. While we are still early in the evaluation of this Architecture and authoring tool set, indications are that we are on the right track. We were able to bring in student interns without prior expertise in authoring dialogue characters and have them build 3 new characters, in addition to our Hassan character testbed.

6. ANALYSIS

In the previous three sections we have seen three different architectures for implementing a question answering character. Which one is best to use? The answer depends on the precise requirements of the domain, as discussed in Section 1. For simple characters who do not need extended negotiations or dynamic mood and emotional models, Architecture 1 may be sufficient. On the other hand, if more extended sequences of interaction are required, and/or the domain will undergo significant amounts of revision, Architecture 3 may be preferable. It is not clear whether Architecture 2 is dominant for any specific requirement set, but it does have advantages over architecture 1 in global coherence, and has less complex authoring demands than architecture 3.

7. CONCLUSIONS

In this paper we have described the tactical questioning training genre for virtual characters. We have presented three different architectures to approach the problem and analyzed their strengths and weaknesses. These three architectures are by no means a complete set of possible approaches. There are many ways in which they may be extended in coverage and performance. Still, comparing the same character built in three different ways is instructive as to the specific advantages and disadvantages that each architecture provides.

ACKNOWLEDGMENTS

The project or effort described here has been sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

REFERENCES

[Army, 2006] Army. Police intelligence operations: Appendix d: Tactical questioning. Technical Report FM 3-19.50, Department of the Army, 2006. retrieved from https://rdl.train.army.mil/soldierPortal/atia/adlsc/view/public/22739-1/FM/3-19.50/pref.htm.

[Artstein *et al.*, 2008] Ron Artstein, Sudeep Gandhe, Anton Leuski, and David Traum. Field testing of an interactive question-answering character. In *ELRA Workshop on Evaluation*, Marrakech, Morocco, May 2008.

[Busso *et al.*, 2008] Carlos Busso, Sungbok Lee, and Shrikanth Narayanan. Analysis of emotionally salient aspects of fundamental frequency for emotion detection. *IEEE Transactions on Speech and Audio and Language Processing*, 2008. to appear.

[Devault *et al.*, 2008] David Devault, David Traum, and Ron Artstein. Making grammar-based generation easier to deploy in dialogue systems. In *Fifth INLG Conference*, 2008.

[Gandhe *et al.*, 2008] Sudeep Gandhe, David DeVault, Antonio Roque, Bilyana Martinovski, Ron Artstein, Anton Leuski, Jillian Gerten, and David Traum. From domain specification to virtual humans: An integrated approach to authoring tactical questioning characters. In *Proceedings of Interspeech Conference*, 2008.

[Kenny *et al.*, 2007a] P. Kenny, A. Hartholt, J. Gratch, W. Swartout, D. Traum, S. Marsella, and D. Piepol. Building interactive virtual humans for training environments. In *Proceedings of I/ITSEC*, 2007.

[Kenny *et al.*, 2007b] Patrick G. Kenny, Thomas D. Parsons, Jonathan Gratch, Anton Leuski, and Albert A. Rizzo. Virtual patients for clinical therapist skills training. In Catherine Pelachaud, Jean-Claude Martin, Elisabeth André, Gérard Chollet, Kostas Karpouzis, and Danielle Pelé, editors, *IVA*, volume 4722 of *Lecture Notes in Computer Science*, pages 197–210. Springer, 2007.

[Kenny *et al.*, 2008] Patrick Kenny, Thomas D. Parsons, Jonathan Gratch, and Albert A. Rizzo. Evaluation of justina: A virtual patient with ptsd. In Helmut Prendinger, James C. Lester, and Mitsuru Ishizuka, editors, *IVA*, volume 5208 of *Lecture Notes in Computer Science*, pages 394–408. Springer, 2008.

[Lee and Marsella, 2006] Jina Lee and Stacy Marsella. Nonverbal behavior generator for embodied conversational agents. In Jonathan Gratch, Michael Young, Ruth Aylett, Daniel Ballin, and Patrick Olivier, editors, *IVA*, pages 243–255. Springer, 2006.

[Leuski *et al.*, 2006a] Anton Leuski, Brandon Kennedy, Ronakkumar Patel, and David Traum. Asking questions to limited domain virtual characters: How good does speech recognition have to be? In *25th Army Science Conference*, 2006.

[Leuski *et al.*, 2006b] Anton Leuski, Jarrell Pair, David Traum, Peter J. McNerney, Panayiotis Georgiou, and Ronakkumar Patel. How to talk to a hologram. In Ernest Edmonds, Doug Riecken, Cécile L. Paris, and Candace L. Sidner, editors, *Proceedings of the 11th international conference on Intelligent user interfaces (IUI'06)*, pages 360–362, Sydney, Australia, 2006. ACM Press New York, NY, USA.

[Leuski *et al.*, 2006c] Anton Leuski, Ronakkumar Patel, David Traum, and Brandon Kennedy. Building effective question answering characters. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pages 18–27, 2006.

[Pellom, 2001] Bryan Pellom. Sonic: The university of colorado continuous speech recognizer. In *University of Colorado, tech report #TR-CSLR-2001-01*, Boulder, Colorado, 2001.

[Roque and Traum, 2007] Antonio Roque and David Traum. A model of compliance and emotion for potentially adversarial dialogue agents. In *In Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, 2007.

[Roque and Traum, 2008] Antonio Roque and David Traum. Degrees of groundedness based on evidence of understanding. In *The 9th SIGdial Workshop on Discourse and Dialogue*, 2008.

[Sethy *et al.*, 2006] Abhinav Sethy, Panayiotis Georgiou, and Shrikanth Narayanan. Selecting relevant text subsets from webdata for building topic specific language models. In Proceedings of HLT-NAACL, 2006.

[Thiebaux *et al.*, 2008] Marcus Thiebaux, Andrew Marshall, Stacy Marsella, and Marcelo Kallmann. Smartbody: Behavior realization for embodied conversational agents. In *Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2008.

[Traum *et al.*, 2007] David Traum, Antonio Roque, Anton Leuski, Panayiotis Georgiou, Jillian Gerten, Bilyana Martinovski, Shrikanth Narayanan, Susan Robinson, and Ashish Vaswani. Hassan: A virtual human for tactical questioning. In *The 8th SIGdial Workshop on Discourse and Dialogue*, 2007.