Challenges in Interpreting Spoken Military Commands and Tutoring Session Responses

Elizabeth Owen Bratt, Karl Schultz, and Stanley Peters

Stanford University CSLI

Proceedings of the GEAF 2007 Workshop
Tracy Holloway King and Emily M. Bender (Editors)

CSLI Studies in Computational Linguistics ONLINE

Ann Copestake (Series Editor)

2007

CSLI Publications

http://csli-publications.stanford.edu/

Abstract

Various challenges have emerged over several years of grammar engineering for the spoken dialogue interface to the Navy damage control simulator DC-Train and the Spoken Conversational Tutor SCoT-DC, which reviews DC-Train performance. The systems use two methods for finding interpretations for student utterances from the recognized string. First, a Gemini grammar interprets full strings into a complex, structured logical form. A successful Gemini logical form is the preferred interpretation. Next, a robust Nuance natural language grammar looks for any interpretable phrases in the utterances which Gemini could not interpret, and uses heuristics to determine the best set of slots and values. Voice-Enabled DC-Train and SCoT-DC face challenges due to speech recognition errors, disfluent speech, underspecified responses requiring dialogue context for disambiguation, students' varying levels of familiarity and skill at using the preferred military terminology, and the need for coordination with the dialogue manager's strategies for clarification, confirmation and modeling of student uncertainty.

1 Introduction

We examine various types of challenges faced during the grammar development for the spoken language interfaces to the DC-Train Navy damage control simulator (Bulitko & Wilkins 1999) and to the Spoken Conversational Tutor (SCoT-DC) (Schultz et al. 2003), which reviews a student's DC-Train performance. First, we describe the task of damage control and the types of utterances which require coverage. Next, we present an overview of the spoken natural language architecture and systems used. Finally, we review various categories of spoken input that required special strategies, constraints or decisions during grammar engineering.

2 Spoken Commands and Discussion in the Damage Control Domain

Voice-enabled DC-Train (VE-DCT) provides a student the opportunity to play the role of the Damage Control Assistant (DCA) on a DDG-51 destroyer. As DCA, the student is responsible for receiving messages from others on the ship and making decisions about which personnel should take which actions to combat adverse events like fires, flooding, and smoke in any of the 484 compartments on the ship.

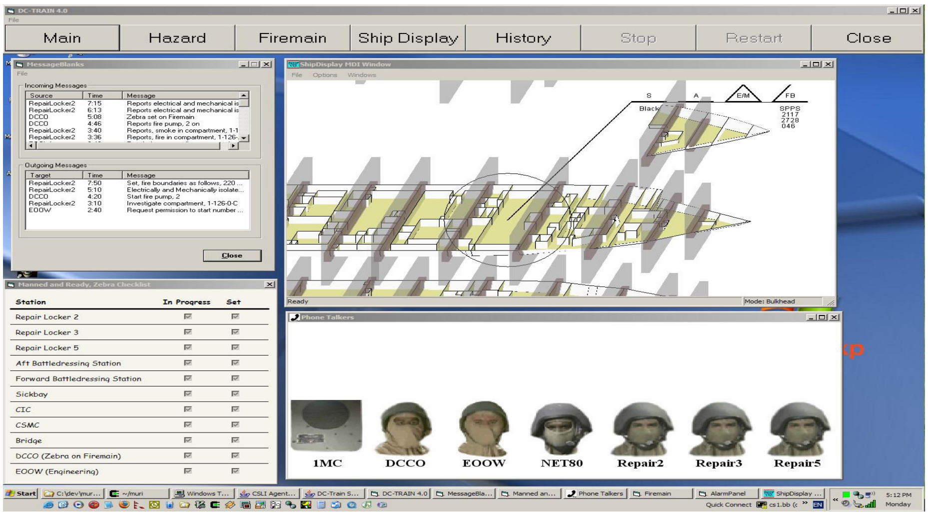


Figure 1. DC-Train Information Windows

The simpler forms of speech to VE-DCT include brief acknowledgements of incoming messages (*affirmative* or *DCA aye*) and brief cancellations of incorrect commands (*cancel that* or *negative*).

In their full form, commands to personnel always involve identification of the addressee. Full commands also identify either the speaker (*DCA*) or the method of communication to the addressee (e.g. *NET80*, for broadcast throughout the ship). Next, full commands contain the desired action and any required parameters, such as a boundary (e.g. *primary aft 97*), a compartment number (e.g. *2-126-2-C*) or a compartment name (e.g. *Combat System Equipment Room Number 2*). This results in commands such as *NET80 to Repair Two electrically and mechanically isolate compartment 1-126-0-C* and *Repair Two, DCA, set smoke boundaries primary forward 42 primary aft 78 secondary forward 18 secondary aft 97 above 1 below 3.* Requests for permission (*EOOW, DCA, request permission to start fire pump number two*) and informative communication (*NET80 to CO, all stations are manned and ready, zebra is set*) are similar in form to commands.

VE-DCT also allows the student to omit the addressee (along with the speaker or method of communication) and any parameters of a command, as long as the student fills in the required parameters in response to system queries. This multi-turn method of issuing commands takes place as seen in Figure 2.

Student: set smoke boundaries
VE-DCT: DCA interrogative for repair team and boundary bulkheads
Student: repair three
VE-DCT: DCA interrogative for boundary bulkheads
Student: primary forward 42, primary aft 78
VE-DCT: DCA interrogative for secondary boundaries
Student: secondary forward 18, secondary aft 97
VE-DCT: To Repair Locker 3, set smoke boundaries as follows: 97, 78, 42, 18, Aye

Figure 2. Interrogative Dialogue in VE-DCT

In SCoT, the student reviews events from the VE-DCT session with an automated tutor, discussing in particular areas where the student did not take required actions in the correct order, the correct selection of repair team by region of the ship, the correct selection of bulkheads to set boundaries for preventing the spread of fire, smoke or flooding, and how to prioritize actions by the type and location of compartments affected.

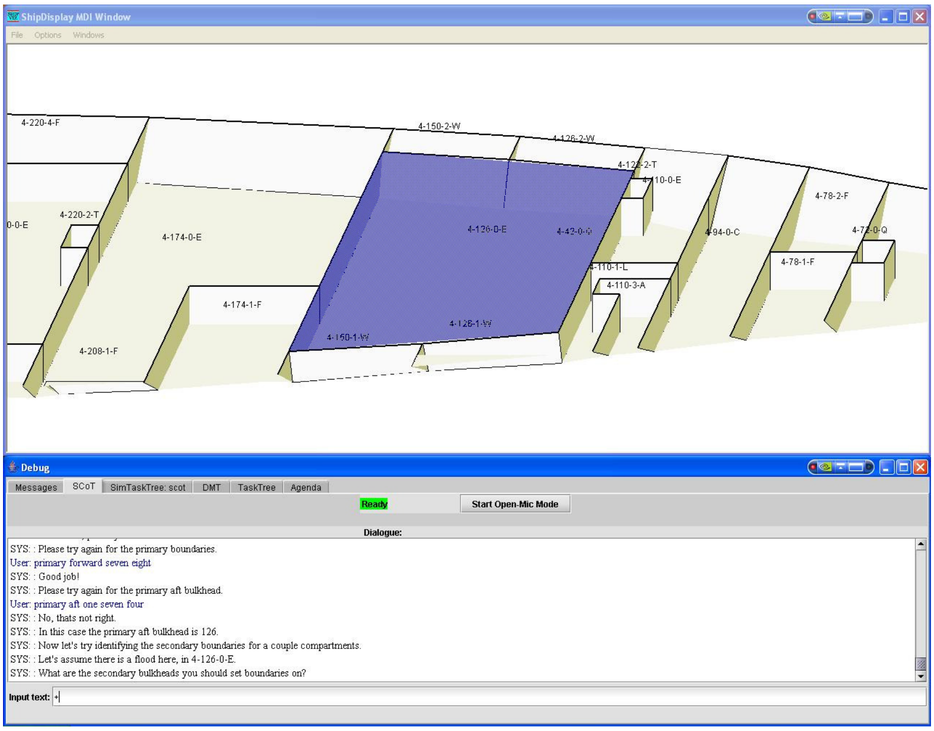


Figure 3. SCoT Display

Since SCoT takes the initiative in leading the tutoring session, the student's utterances are generally responses to tutoring questions, such as

Assuming you have a fire report, there are 2 other things you should have done before ordering fire fighting. Lets begin with the first 1. What is it? The student responses can be very brief (e.g. *investigate*) or take a longer form (e.g. *I guess I should um set fire boundaries first.*) SCoT also permits gestural input in response to some questions, such as clicking on a compartment or circling it on the ship display. SCoT does not have any tutorial discussions in which it would be natural for the student to speak and click or circle compartments at the same time, so natural language interpretation has not had to support gestural constraints, though this capability would be supported by the CSLI dialogue manager architecture (Lemon et al. 2001, Schultz et al. 2003).

In addition to the questions to which students often answer with a short, simple verb or noun phrase, SCoT also asks more open-ended questions, requesting definitions for terms (e.g. *First of all can you tell me what primary boundaries are?)* and reasons for actions (e.g. *Why is it necessary to isolate when you have a report of fire?*). Answers to these questions generally are syntactically more complex, such as *First two bulkheads around the crisis*, *To see if it's a false fire*, and *Prevent smoke from spreading to other compartments*.

VE-DCT and SCoT have been used in a number of experiments and data collections, which have given results for the experimental conditions studied, but also on the range of possible user input and ways it can support student modeling (Jones, Bratt & Schultz 2007).

Theme	# of Subjects	Type of Subject	Year	Results
Different Tutoring Topics at Different Times	30	Stanford students	2004	Pon-Barry et al. 2004, Peters et al. 2004
Natural Language-based Tutoring Strategies	40	Stanford students	2004	Pon-Barry et al. 2006
Multimodality and Active/Passive Tutoring	210	U.S. Naval Academy midshipmen	2005	Bratt et al. 2005
Human Coaching with DC-Train and SCoT	5	Stanford students	2005	Bratt et al. 2005
Human Coaching with DC-Train and SCoT	10	Surface Warfare Officers' School students	2006	No paper yet

Figure 4. Experiments with DC-Train and SCoT

3 Architecture of the Natural Language Interface

After the spoken input into VE-DCT and SCoT is transformed into a string of words by Nuance speech recognition, the natural language understanding takes place in one of two components, as shown in Figure 5.

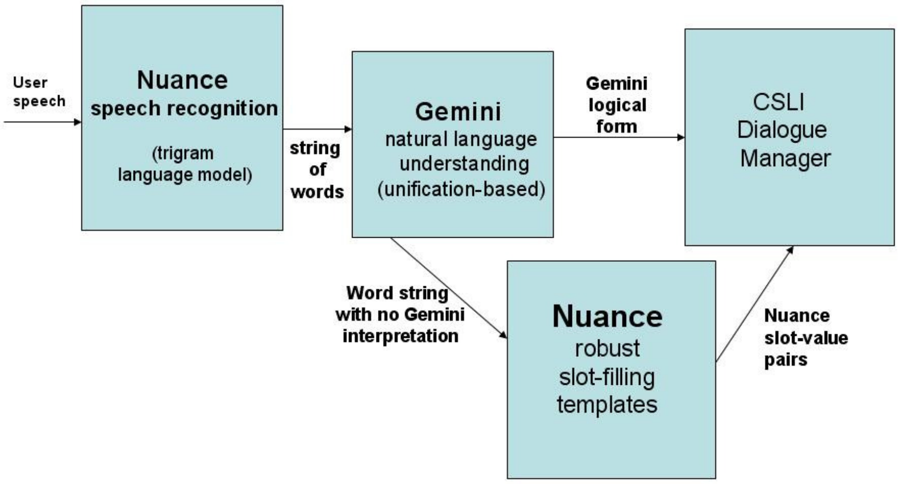


Figure 5. Paths of Interpretation

If the string is well-formed in our Gemini unification-based grammar (Dowding et al. 1993), then it receives a Gemini logical form, from which the Dialogue Manager will extract relevant information. If Gemini cannot interpret the string, perhaps because it has disfluencies or previously unencountered phrasings, then a robust slot-filling Nuance grammar will look for the maximum number of words it can interpret with the minimum number of grammar rules, and return slot-value pairs for its best interpretation.

Our model for these two forms of interpretation is that Gemini is intended to be a linguistically motivated grammar, which uses phrases like NP (noun phrase) and VP (verb phrase), and builds a logical form (LF) capable of representing embedding and other complex relationships. The Nuance slots are intended as a fallback mechanism, which capture partial meaning that is helpful when we cannot understand the entire utterance. The Nuance slots are mainly a flat representation, though certain items, such as boundaries do involve nesting of a single level of structure (rank, direction and frame number for each boundary). Because the Nuance slots involve filling slots from phrases, in sentences which did not parse in Gemini, the emphasis is on very local interpretation. This means that very particular, idiosyncratic patterns are easy to include in Nuance, since there is less chance of their affecting rules elsewhere, than in a more interconnected, richer system like Gemini. The Nuance slots are close to the domain representation used by the dialogue manager, so they allow for a quick development cycle, with little effort spent compared to the effort needed for any new complexities in a Gemini logical form.

One characteristic of the Nuance slots which can severely limit their utility for certain kinds of discussions is that they permit only one instance of a slot to be filled per sentence. If there are multiple items of the same kind in the same utterance, Nuance natural language rules must provide separate slots for each item so that all of them can be interpreted individually. Without this provision, the information in the slot would be overwritten by each new phrase that matched the Nuance rule, and the only slot information provided to the dialogue manager would be that of the final eligible phrase. We encountered this situation with boundaries, and we defined slots for the usual number of up to four boundaries within an utterance to account for it. But we do at times run into problems with actions, since usually there is only one per utterance, and if there are two, the later one will overwrite the information from the earlier one. Another area where Nuance's behavior of overwriting later slots gives us trouble is when students issue commands with conjunctions, such as *Set fire and smoke boundaries*, *Fight fire in compartments 1-174-01-L and 3-116-1-T*, or *Set flooding boundaries and electrically and mechanically isolate compartment 3-310-2-L.* Students rarely use conjunctions, so this is not a frequent problem, but it has happened at times. The main place students use conjunctions is for pairs of related boundaries, such as both primary boundaries or the above and below decks for vertical boundaries, and our robust interpretation treats these conjunctions the same as the list of boundaries we expect.

The two layers of interpretation for robustness and confidence, i.e., Gemini first, then Nuance, have served our system fairly well. In recent error analysis, we have considered the possible utility of an additional layer of less reliable Nuance slots, so that a complete Gemini LF would be preferred, then the Nuance slot-values which seem fairly reliable, then if there are none of these, we could use the less reliable Nuance slots to start a clarification dialogue with the user, rather than simply asking the user to repeat the utterance.

Our current model of interpreting Nuance slots requires an "action" slot to be filled, indicating what kind of command is intended, and allows the parameters to be filled in later. However, if a set of parameters slots are filled, but the action is not, we might consider clarifying with the user if a particular action was intended. One reason for our current requirement of an "action" slot is that many parameters can involve numbers as values, and many numbers are short words (e.g. *two* or *eight*) which can result from misrecognitions. Thus it is very easy to have a misrecognized sentence which appears to have parameter values of short numbers, when this is not actually the case. If we had less reliable, back-off Nuance slots, we might have those slots only filled by numbers with identifiers such as *frame number*, *pump number*, etc., as opposed to the numbers which occur in a more standard, full command.

The perspective of what the dialogue manager will do with an interpretation is important to keep in mind. If the dialogue manager requires an "action" slot to be able to clarify a partially understood command, it is particularly important to make the interpretation rules yielding action slots robust, such as making sure they work if there are filled pauses in likely places, or other possible word variations.

Our Gemini grammar currently includes 183 grammar rules, 949 one-word lexical entries, and 2991 multi-word lexical entries. The vocabulary includes 51 action verbs (some synonymous), 36 lexical items for ship personnel, 396 compartment names, 2152 frame numbers (for compartments, bulkheads, valves, etc.), and 23 synonyms for *yes*. In earlier versions of our system, we compiled the Gemini grammar into a language model for Nuance speech recognition (Moore 1998). This kept our coverage for speech recognition and natural language understanding tightly synchronized, and allowed development for the natural language understanding to automatically produce speech recognition results; however, as our system coverage grew, we experienced various problems which led us to abandon this approach. Specifically, compiling the Gemini grammar into a Nuance grammar took a long time, as long as a day, which made it difficult to address bugs or test out alternate options rapidly. Also, as the grammar grew more complex, speech recognition began to get significantly slower than real time, which made it less practical for a dialogue system. Finally, certain Gemini grammars would produce Nuance grammar files that would fail either at Nuance compile time or Nuance run time without explanation.

Currently, we train a trigram model on a corpus of utterances from our past experiments, using Nuance's standard tools to create a probabilistic finite state grammar. The entire process is much faster, and can be completed in under 15 minutes, so we have a more responsive development cycle. Using trigram recognition has improved our speech performance by making us more robust to unanticipated phrasings and out of vocabulary words. We currently train a DC-Train language model on a corpus with 11,551 utterances, containing 390 unique words, and a SCoT language model on a corpus with 19,123 utterances, containing 1780 unique words. The SCoT corpus has more unique words than the DCT corpus because in addition to the words for the basic commands to the simulator, SCoT involves discussion of why to take certain actions, definitions for terms, and discussion of which kinds of compartments should be prioritized over others.

3.2 Tools to Support Grammar Development

In order to support grammar development, as well as speech recognition language model development, we have developed a number of tools. Our most powerful tool is our Nuance Batchrec Analysis Tool, batchdb.[1] Batchdb reads in results of a Nuance batch recognition run, calculates performance metrics (such as word error rate), and stores the results in a MySQL database. With batchdb, we can use SQL queries to compare data from batch runs with different language models or different Nuance parameter settings. We look to optimize for the lowest word error rate, the most accurate semantic slots, and recognition run-time under real time.

Batchdb enters various different views of the data automatically into the database, so that it is easy to work with the exact representation needed for the task at hand. For example, our transcriptions include annotations within square brackets such as *[annoyed]* or *[pause]*. For some data analysis, we are interested in those annotations; for others, we only want to see the transcribed words. Similarly, in some data views we use expletives transcribed as they were spoken, but when we use data with expletives in the language model we sanitized them all into the single word *expletive*, which we give all its various possible pronunciations, so that users of the system will not see any actual expletives on the system display, though our capability to recognize them helps us deal with the full range of military speech!

Batchdb also automatically standardizes the spelling in the transcriptions. We have set forth policies on how transcribers are to enter words, but we have found that it can be hard for transcribers to keep them in mind, so it is good to have automatic processing to eliminate spurious distinctions such as *ok* vs. *okay*, and versions of Navy words with the individual letters treated as distinct words, e.g. *d. c. c. o.* (for *Damage Control Console Operator*), vs. our preferred treatment of them as a single word, e.g. *dcco*.

Batchdb also pre-processes the data before running the sclite speech recognition scoring program. Sclite allows variant words to be scored as equivalent. We have instances of homophones which are sometimes able to be disambiguated by sentence context and other times not. For example, *two four* is right, while *two fore* is wrong, but *fore* vs. *four* in an isolated utterance are equivalent. We keep the versions of the word distinct in our transcript, where the dialogue context can distinguish them, but when we score our speech recognition using a single language model for all contexts, we are able to penalize errors that the language model could detect while overlooking cases of homophone mismatch that speech recognition has no capability to deal with. (Our dialogue manager can automatically correct for these cases.)

---

[1] Available to the public under the GPL open-source license at http://sourceforge.net/projects/nuance-batchrec .

Other tools we use to support our development are a Transcriber Wavefile Preparation Tool, a perl script which takes individual wavefiles from a dialogue and creates a session wavefile and transcript file for Transcriber software (Barras et al. 2000). We use a Transcript Tracker web application to manage transcription file status and coordinate between transcribers, so that they do not duplicate effort, and to provide us a means for automatically standardizing variants or calling the transcriber's attention to items that should be fixed before the transcription is complete.

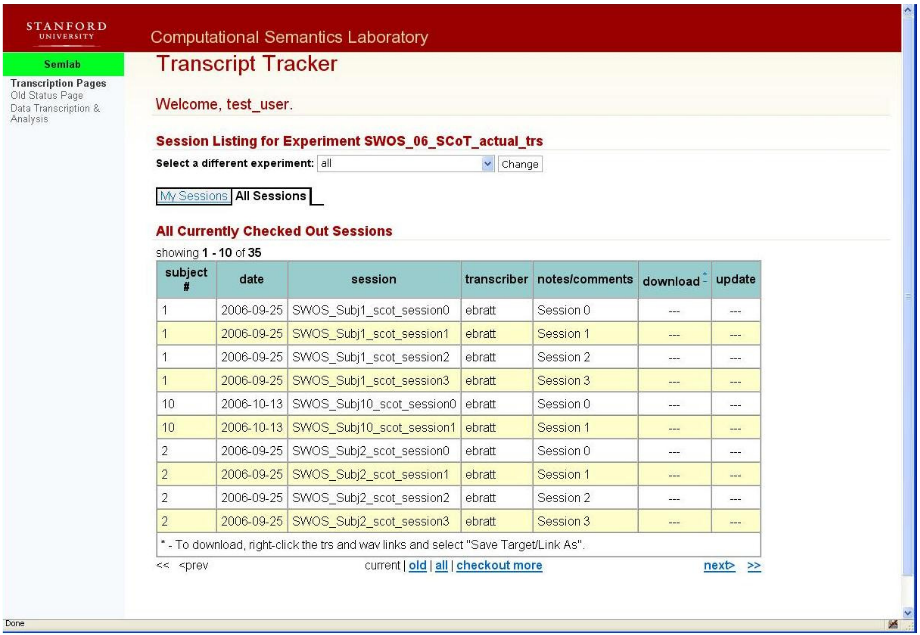
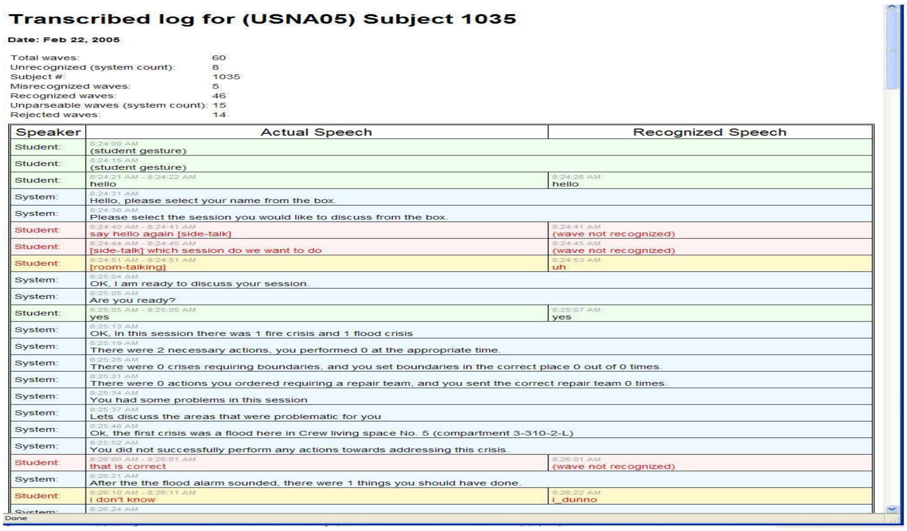


Figure 6: Transcript Tracker

We also use a Transcript Preparation Tool, which combines the automatically logged system transcript with transcriptions from Transcriber .trs files for html viewing, to help us review system performance at a dialogue level. Figure 7 shows an example of the XML-based transcripts produced.



Transcribed log for (USNA05) Subject 1035

Date: Feb 22, 2005

Total waves:	60
Unrecognized (system count):	8
Subject #:	1035
Misrecognized waves:	5
Recognized waves:	46
Unparseable waves (system count):	15
Rejected waves:	14

Speaker	Actual Speech	Recognized Speech
Student:	8:24:00 AM (student gesture)	
Student:	8:24:15 AM (student gesture)	
Student:	8:24:21 AM - 8:24:22 AM hello	8:24:26 AM hello
System:	8:24:31 AM Hello, please select your name from the box.	
System:	8:24:38 AM Please select the session you would like to discuss from the box.	
Student:	8:24:40 AM - 8:24:41 AM say hello again [side-talk]	8:24:41 AM (wave not recognized)
Student:	8:24:44 AM - 8:24:45 AM [side-talk] which session do we want to do	8:24:45 AM (wave not recognized)
Student:	8:24:51 AM - 8:24:51 AM [room-talking]	8:24:53 AM uh
System:	8:25:04 AM OK, I am ready to discuss your session.	
System:	8:25:05 AM Are you ready?	
Student:	8:25:05 AM - 8:25:05 AM yes	8:25:07 AM yes
System:	8:25:13 AM OK, In this session there was 1 fire crisis and 1 flood crisis	
System:	8:25:19 AM There were 2 necessary actions, you performed 0 at the appropriate time.	
System:	8:25:25 AM There were 0 crises requiring boundaries, and you set boundaries in the correct place 0 out of 0 times.	
System:	8:25:31 AM There were 0 actions you ordered requiring a repair team, and you sent the correct repair team 0 times.	
System:	8:25:34 AM You had some problems in this session	
System:	8:25:37 AM Lets discuss the areas that were problematic for you	
System:	8:25:48 AM Ok, the first crisis was a flood here in Crew living space No. 5 (compartment 3-310-2-L)	
System:	8:25:52 AM You did not successfully perform any actions towards addressing this crisis.	
Student:	8:26:00 AM - 8:26:01 AM that is correct	8:26:01 AM (wave not recognized)
System:	8:26:21 AM After the the flood alarm sounded, there were 1 things you should have done.	
Student:	8:26:10 AM - 8:26:11 AM i don't know	8:26:22 AM i_dunno
System:	8:26:24 AM	

Done

Figure 7: XML Transcript of Session

We use a Transcript Search application, based on our Transcript Indexing Tool, which allows a user to search our XML transcripts for instances of words or regular expressions, in modes which include or exclude items such as square-bracketed annotations, word fragments, and punctuation. The search results are displayed as an index web page with links to the dialogues containing the item searched for, as shown in Figure 8. Searching transcripts this way is useful for examining dialogue performance for particular questions, for example.

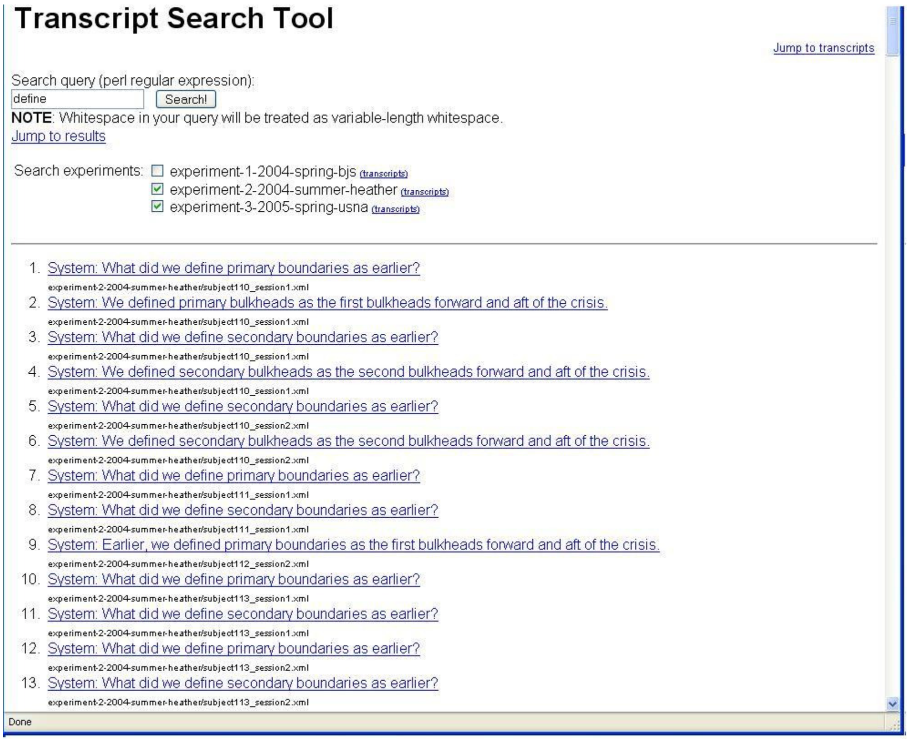


Figure 8: Transcript Search Tool Result Page

4 Interpretation Challenges for Grammar Engineering

4.1 Issues in Training

Because VE-DCT and SCoT are intended to train students, the spoken interfaces need to permit student mistakes. For VE-DCT, the system needs to understand nonexistent compartments and boundary locations, which the student might construct by using the standard format for these items, because simply not recognizing or not understanding the utterance would not help the student realize that the compartment did not exist; rather, it would look like a general system failure. For SCoT, the tutor needs to understand likely wrong answers as well as the correct, expected answer.

Another aspect of training people with a spoken system is how they relate to the tutor or spoken system, and how much they treat it as a person (Nass & Brave 2005, Reeves & Nass 1996). In using our system in a classroom setting at the U.S. Naval Academy, students produced all of the specific examples of uncooperative speech shown in Figure 9, and many other examples besides these. Students who produced uncooperative utterances did not have significantly different DC-Train performance, overall test scores, or learning gains. At the time of our experiment, our tutor did not understand these utterances, and treated them identically to a student utterance of *I don't know*, and moved on. Providing grammar coverage and suitable interpretations or categorizations of this kind of utterance would be a significant additional task for the grammar engineer; however, in a training system used in the real world, this may be a concern worth addressing.

Polite Complaint	*I can't see what compartment you're talking about*
Swearing	*F--- you*
Insult (to the system)	*I hate you*
Threat (to the system)	*I ought to pulverize your guts out*
Mocking	*There you go it only took you four times*
Inappropriate Response	*Have a beer*
Reaction to Reprimand	*I bet you couldn't do any better man*
Intentionally Using Another Language	*Siete*
Generally Antagonistic or Unresponsive	*This is dumb*

Figure 9. Types of Uncooperativity Encountered by SCoT

4.2 Issues with Military Domain

Another challenge for grammar engineering for a system that involves a simulation of a situation the student may be familiar with in reality is that the student may use more domain knowledge than the simulator supports. For example, the student may not only give a *desmoke* command, which our system would support, but then go on to specify that box fans should be used for desmoking, which is beyond the scope of the simulator. Another area where students familiar with actual ships might use their real world knowledge in ways that make grammar engineering more difficult is to use synonyms in complex compartment names, such as *berthing* for *living space* or *Chief* for *CPO*. Dialogue context might also make it natural to omit certain parts of complex compartment names, such as numerical or directional identifiers, like *number one, forward, aft, port*, or *starboard*. The same compartment might be called *CPO Berthing, Chief's Berthing* or *CPO Living Space Number One.*

Another issue in a training system is how to support standard vs. nonstandard terminology and phrasing. For example, a hyphen written between numbers in compartment identifiers is pronounced as *tac*, but a student less familiar with military terms might pronounce it as *dash*, or omit it. Similarly, letters used in compartment names are pronounced according to the Navy's phonetic alphabet, e.g. *Charlie*, whereas someone unfamiliar with the convention might just pronounce the letter *c*. Another Navy pronunciation convention is to spell out each digit of a number individually, to minimize misunderstandings such as *fifty* vs. *fifteen*. Thus, the standard pronunciation of *220* would be *two two zero*, not *two hundred twenty*. For a compartment number such as *1-126-0-C*, in addition to the standard pronunciation of *one tac one two six tac zero tac charlie*, there are many possible non-standard pronunciations, such as *one dash one hundred twenty six dash oh dash c*, *one tac one twenty six tac zero tac c*, and so on. We give users an introduction to our system which explains how to pronounce these items, and military users are familiar with these conventions, so most users will produce the standard pronunciations. However, sometimes they will produce the non-standard pronunciations. The challenge is to recognize those times correctly, while not having the non-standard options overwhelm the correct ones, and lead to misrecognitions. Another challenge is the degree to which to represent the non-standard choices in the interpretation, as material for the tutor or coach to comment on.

A similar issue in interpretation in a training dialogue system is how much to assume from the student's wording, how much to clarify, and how to convey to the student that a wording is not ideal. For certain commands, the correct method of delivering them is to include specifications of exactly the relevant areas the commands apply to, or the precise type of action they mean. For example, in the context of the DC-Train simulator, students should be giving the command *electrically and mechanically isolate*, not just the command *isolate*. Similarly, they should give one of the commands *Set fire boundaries*, *Set smoke boundaries*, or *Set flooding boundaries*, not the command *Set boundaries* without specifying which kind. Again, it becomes a question of how much to support the non-standard variants, which do occur, without biasing the system to accept them too much, and how to let the student know not to use them. The simple answer of having the system not understand non-standard variants generally just leads to frustration and the perception of the system as failing, rather than calling attention to the fine points of difference in the student's answer.

Another area of interaction of requiring the student to give a proper full specification and what might in reality be identifiable from dialogue context is the military doctrine that commands should identify the addressee and speaker, or communications system to be used for the addressee. Thus, a proper, full command with the addressee *repair three* and speaker *DCA* would be *Repair three, DCA, investigate compartment two tac two two zero tac oh one tac lima*. A proper full command instructing a phone talker (communications assistant in damage control central) to use the Net-80 communications system, which broadcasts throughout the ship, rather than a limited radio address only to the specific addressee (in this case the CO, or ship's captain), would be *Net eighty to CO, all stations manned and ready, zebra is set*.

A rich domain can also produce a wide range of possible answers if questions are open-ended. In our experiments at Stanford, USNA, and SWOS, we asked students to provide definitions of terms and to explain why certain actions should be taken. Fortunately for us, the answers students gave ended up being reasonably tractable for interpretation. In the Stanford confidence-sensitivity experiment (2004) and the USNA experiment (2005), the mean utterance length for the 725 answers to open-ended questions was 5.17 words.

4.3 Issues with Spoken Dialogue Context

Because VE-DCT and SCoT involve a system using natural language for its interactions with the student, specifically, speaking to the student in a synthesized voice (Festival for VE-DCT, Taylor et al. 1998, and FestVox limited domain voice for SCoT, Black and Lenzo 2003), grammar development has to account for the possibility that the student is likely to use vocabulary or phrases used by the system. Development of system output must be coordinated with development of the speech recognition language model and the interpretation grammar.

In a system with various numerical parameters, and a series of commands and spoken interactions that create a dialogue context, it becomes natural for the student to use numbers without modifiers or units. The representation of the meaning of these numbers has to allow for their ultimate interpretation by the dialogue manager as specifying the correct kind of item, and not specifying something incorrect. When our system involves the dialogue context of the system issuing an "interrogative" request to the user for missing parameters, we have interpreted the numbers as filling the specific parameters we need, and have ignored the possibility of their filling other parameters. The grammar produces multiple interpretations for bare numbers, such as *two*, which can be interpreted as a pump number, a repair party identifier, a frame number or a deck number. The dialogue manager decides which type of number is likely, and interprets the number accordingly. An alternate method of this would be to have distinct grammars which the system switches between based on dialogue context. In either approach, the dialogue manager must control the interpretation, either by choosing a narrower grammar or by choosing from information provided by the grammar. The complex logical form constructed by Gemini lends itself more naturally to the approach where the dialogue manager chooses the grammar in advance, or an approach where the grammar provides an underspecified number which the dialogue manager adds type information to. The Nuance slot approach can give an underspecified number, but it can also give a series of unrelated slots, which provide more information than can possibly all be true at the same time. This approach allows the grammar to use any constraints from wording that might be possible to give the dialogue manager an exact set of possible ways to interpret the underspecified utterance. Our Gemini grammar currently gives a set of different interpretations, of which only one is provided to the dialogue manager. If that one does not match the kind of information expected in the dialogue context, the dialogue manager checks the Nuance interpretation for a slot matching the information it expected. It ignores all slots present that are not what it is looking for. The Nuance interpretations involve several different slots, and the dialogue manager adds in several possible slots through reasoning.

4.4 Issues with Speech Input

Our speech recognition word error rate has tended to be around 8% in most of our experiments, though it varies by speaker. Our rejection rate, i.e. the percentage of sentences which our recognizer cannot produce a hypothesized string of words it is sufficiently confident in, is around 4%.

Spoken input can lead to a number of challenges for a grammar, as long as the recognition possibilities are more than what is covered by the grammar. If the utterances involve disfluencies or out of vocabulary items, the recognized string will be an unsuccessful attempt to match what the speaker actually said, so the various places where the grammar backs off to acoustically similar items might produce unexpected results. Acoustic similarity between words can also produce unanticipated strings for the grammar writer to capture, such as *blue* being misrecognized for *below* in our data, as well as other confusion pairs such as *l* and *alpha*, or *set* and *send*. Thus, interpretation rules are more robust when they allow for these.

A general issue for all grammar coverage is dealing with unexpected phrases, though in spoken input the fact that the phrases pass through a speech recognizer first may introduce additional problems if the resulting string does not match what the speaker says.

A disfluency involving repeated words or filled pauses (*um*, *uh*) might present difficulties for a grammar expecting particular phrases with particular constituents without interruptions. Incomplete sentences present another variant of this problem, when a student either is cut off by the speech recognizer endpointer by pausing too long, or the student actually stops speaking part-way through a command. For example, 8% of utterances in the USNA corpus that involve a student beginning to give a compartment number are broken off before the compartment number is complete, and 13% of utterances with a repair party identified break off before giving the repair team their command. It would be desirable to interpret the material actually said, but the phrases will not be complete. Noting where the student breaks off may also help model that this kind of information may be difficult for the student to figure out (Jones, Bratt & Schultz 2007).

Another area in which the spoken input interacts with the grammar interpretation is when setting the word transition weight parameter for the speech recognizer. This parameter can be set so that the speech recognizer is discouraged from hypothesizing short words to match the acoustics of the speech signal in favor of interpreting the same material as part of existing words. This can interact with grammar interpretation of variant formulations of utterances. In our 2006 data collection at SWOS, a student said to the system *Repair repair five send investigators to engine room number two*, which we recognized as *Repair five investigate niner two engine room number two*. Our grammar covered *investigate*, but not *send investigators to*, so we were fortunate that the speech recognition chose a hypothesis that caught the main word and not the more verbose variant form of the command.

Another point about spoken input is that the speech channel allows the user to provide more information than just the straight content of the words (Pon-Barry et al. 2006), as they would if they were choosing a command or an item from a menu. Locations of pauses before items (Jones, Bratt, & Schultz 2007), or the fact that nonstandard vocabulary has been used, can signal additional information that a tutoring system can use to better model what the student knows well and what the student may be struggling with. The interpretation of an utterance given by a grammar can either contain this kind of information, e.g. for nonstandard phrasing, or provide a representation which helps support information from separate processing, as in pause detection.

5 Conclusions

Data from users of VE-DCT and SCoT over the course of multiple experiments provide us with challenges for grammar engineering in various of the complex aspects of our training system: training, representing a complex military domain, recognizing speech, and interpreting speech in dialogue context. Our system architecture provides a preference for canonical input that can be given a complete, structured interpretation, but allows a fallback to robust interpretation of phrases providing partial information. Various system development tools help us detect problems and gauge the success of the solutions we implement.

Acknowledgements

This work is supported by the Department of the Navy under research grant N000140010660, a multidisciplinary university research initiative on natural language interaction with intelligent tutoring systems and research grant N000140510144, on Spoken Language Coaching During Dynamic Problem Solving. We are grateful to Perry McDowell of the Naval Postgraduate School, and Lt. Jonathan Hopkins and Lt. Tyson Young of the Surface Warfare Officers School for providing valuable information about the language of the damage control domain. We are responsible for any errors in our use of their information.

References

Barras, Claude, Geoffrois, Edouard, Wu, Zhibiao, and Liberman, Mark. 2000. Transcriber: development and use of a tool for assisting speech corpora production. *Speech Communication special issue on Speech Annotation and Corpus Tools*, Vol 33, No 1-2, January 2000.

Black, Alan W. and Lenzo, Kevin A. 2003. Building Synthetic Voices for FestVox 2.0 Edition. Available at http://www.festvox.org/bsv/

Bulitko, Vadim V., & Wilkins., David C. 1999. Automated instructor assistant for ship damage control. In *Proceedings of AAAI-99*.

Dowding, John., Gawron, Jean Mark, Appelt, Doug, Cherny, Lynn, Moore, Robert and Moran, Doug. 1993. *Gemini: A Natural Language System for Spoken Language Understanding*. In the Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, Columbus, OH.

Jones, Bevan, Bratt, Elizabeth Owen, and Schultz, Karl. 2007. The Prosody of Uncertainty for Spoken Dialogue Intelligent Tutoring Systems. Manuscript in preparation.

Lemon, Oliver, Anne Bracy, Alexander Gruenstein, and Stanley Peters. 2001. The WITAS Multi-Modal Dialogue System I. In *Proceedings of EuroSpeech 2001*.

Moore, Robert. 1998. Using natural language knowledge sources in speech recognition. In *Proceedings of the NATO Advanced Studies Institute*.

Nass, Clifford. and Brave, Scott. 2005. *Wired for speech: How voice activates and advances the human-computer relationship*. Cambridge, MA: MIT Press.

Pon-Barry, Heather, Schultz, Karl, Bratt, Elizabeth Owen, Clark, Brady, and Peters, Stanley. 2006. Responding to Student Uncertainty in Spoken Tutorial Dialogue Systems. In *International Journal of Artificial Intelligence in Education (IJAIED)* Volume 16, 171-194. Special Issue "Best of ITS 2004" (editors James Lester, Rosa Maria Vicari and Fabio Paraguaçu).

Reeves, Byron. and Nass, Clifford. 1996. *The media equation: How people treat computers, televeision, and new media like real people and places*. New York: Cambridge University Press/CSLI.

Schultz, Karl, Bratt, Elizabeth Owen, Clark, Brady; Peters, Stanley, Pon-Barry, Heather, and Treeratpituk, Pucktada. 2003. A Scalable, Reusable Spoken Conversational Tutor: SCoT. In *AIED 2003 Supplementary Proceedings*, University of Sydney. 367-377.

Taylor, Paul A., Black, Alan, and Richard Caley. 1998. The architecture of the Festival speech synthesis system. In *The Third ESCA Workshop in Speech Synthesis*, pages 147-151, Jenolan Caves, Australia.

U.S. Navy Guide to the phonetic alphabet. http://www.history.navy.mil/faqs/faq101-1.htm