The Vocabulary Problem in Spoken Dialogue Systems

Susan E. Brennan
Department of Psychology
State University of New York
Stony Brook, NY 11794-2500

1. Introduction

Designers of spoken dialogue systems need to be able to predict and constrain the words people use in speech directed at these systems. Larger vocabularies lead to longer processing times, as well as a substantial increase in perplexity errors; according to one estimate (Makhoul 1993), error rates increase with the square root of the number of words in the vocabulary (assuming all words are equally likely). The problem is that a speaker of English may know more than 100,000 words.[1] With such abundance in the mental lexicon, the potential for variability in a speaker's word choices is enormous. Consider, for example, the abstract geometric object in Figure 1 and the referring expressions for it that were spontaneously produced in thirteen different conversations. This object is not lexicalized–it has no conventional label–and in virtually every conversation, speakers took quite a different perspective on it. The mapping of terms to referents is many-to-one, especially when a domain is unfamiliar or when alternative conceptualizations are available. This potential for variability in word choice has been dubbed *the vocabulary problem* by Furnas, Landauer, Gomez, and Dumais (1983 1987).

<<Figure 1 >>

What about word choices in simpler, more familiar domains? We know from studies of naming and categorization that speakers often use basic level terms (such as *chair, shoe,* and *dog)* to refer to common objects. Such terms categorize objects at the maximally informative, basic level (Rosch et al. 1976), rather than at the superordinate level (e.g., *furniture*, *clothing*, and *animal*) or subordinate level (e.g., *recliner*, *loafer*, and *terrier*). Basic level terms are short, easily available in memory, rapid to produce, learned early by children, and highly ranked in word frequency tables. Indeed, in experiments when pairs of people first referred to common objects that were unique in their basic level categories (a shoe, a dog, a car, and a fish), there was much less variability than with the abstract object in Figure 1; speakers used basic level terms 77-89% of the time (Brennan and Clark 1996).

Word choices are also guided by Grice's Maxim of Quantity (Grice 1975), which specifies that speakers should be informative enough, but not too informative. Indeed, the simplest account of referring states that speakers design expressions to distinguish a referent from a set of alternatives, using the most concise expression

that will enable an addressee to pick out the referent uniquely (Brown 1958; Olson 1970). This account was modified by Cruse (1977), whose proposal takes into account not only informativeness but also the enhanced availability of basic level terms. On her proposal, the most *common* term that is also informative enough is considered to be unmarked; all other terms are marked. This explains why speakers are more likely to choose "dog" than "animal" to refer to a terrier, even when the set includes no other living creatures.

This chapter focuses on the vocabulary problem and how it is best managed in human-human as well as in human-computer interaction, and in direct manipulation interfaces as well as in text and spoken dialogue. There are at least two reasons to consider direct manipulation when thinking about spoken dialogue systems. First, these two dialogue styles actually have much more in common than is generally acknowledged (Brennan 1990a). In both, people expect to build up a shared context that enables them to refer to objects introduced or manipulated in previous turns; they expect their partners to keep track of prior context. Both kinds of interfaces rely on symbolic conventions for introducing new information, referring to old (or given) information, and structuring the information. Second, natural language and spoken dialogue systems have been compared unfavorably with direct manipulation systems (see Shneiderman 1886 1992) and have not yet reached their full potential in human-computer interfaces. Sometimes the claim is made that language is inherently ambiguous, with the implication being that icons and other graphical representations are not. I will argue that this is a false assumption; ambiguous mappings can occur in any kind of symbolic interaction, and icons can be just as puzzling as words. Direct manipulation, however, has some built in constraints that do help in managing the vocabulary problem. In order for spoken dialogue systems to better manage the vocabulary problem, system designers need to consider the kinds and sources of variability in word choice in dialogues.

2. The Vocabulary Problem in Dialogues with Computers

In this section I survey some popular approaches to the vocabulary problem in human-computer interaction. These include following the conventions of a specialized group of users, expecting or requiring users to stick to a restricted vocabulary set, giving commands names based on a general measure of word frequency, and allowing systems to use synonyms. Then I consider the vocabulary problem in direct manipulation interfaces, as well as in error messages and other system responses.

2.1. Attempts to Solve the Vocabulary Problem

The traditional approach to labels in command language design, called the *semantic approach* by Katzenberg and Piela (1993), assumes that meanings are invariant across situations. So command language designers should simply choose the best or most conventional labels possible. Yet, as Katzenberg and Piela point out, different situations and working groups have different language conventions, so trying to

identify a single most appropriate convention is a risky enterprise. These researchers suggest what they call a *pragmatic approach*, which takes into account who will be the users of a system and what is the metaphor for its interface. Such an approach has potential when users come from a well-defined group with identifiable conventions (such as, for example, *copyeditors*), but does not solve the vocabulary problem for the generic and diverse populations that many applications aim to serve.

Some have argued that a restricted subset of coverage for English should be sufficient for speech and natural language interfaces, under the assumption that people will use language in a more constrained fashion to computers than to other people. Specifically, the claims are that people addressing computers will restrict themselves to simpler syntax, fewer or no pronouns, and a single, predictable vocabulary of high frequency words (see, e.g., Guindon et al. 1987; Guindon 1991). To support this view, one Wizard of Oz study that simulated a natural language interface to a computerized statistics advisor found that people used very few pronouns (Guindon et al. 1987). However, this study had no control group; the same people might not have used many pronouns to discuss statistics with a human addressee either. In another Wizard of Oz study in a domain where pronouns were more feasible, people used at least as many third-person pronouns to computers as to other people (Brennan 1991). So it appears that pronoun resolution is not an area of coverage that spoken and text dialogue systems can safely ignore.

Unfortunately, the data also fail to support the popular assumption that the vocabulary problem can be solved simply by choosing labels with the highest frequencies. For instance, in studies of the commands people generated for text editing functions, Furnas et al. (1987) found that the likelihood that any two people would produce the same term for the same function ranged from only 7 to 18%. For instance, to remove a file, possible terms include *remove, delete, erase, expunge, kill, omit, destroy, lose, change, rid,* and even *trash*. Some command languages have tried to cope with this problem by accepting synonyms (see, for instance, Good et al. 1984). But as Furnas et al. (1987) discovered in their studies, even with as many as 20 synonyms for a single function, the likelihood of people generating terms from the synonym set for a given function was only about 80% (Furnas et al. 1987). Furthermore, allowing synonyms led to additional problems: In a text editor with only 25 commands, the likelihood that two people who chose the same term actually meant the same function by it was only 15%.

The vocabulary problem abounds in popular software applications; it is all too easy to find examples of puzzling mappings of terms to functions. For instance, in Adobe Photoshop, the basic command for digitizing an image is *acquire*, which seems to represent the application's perspective rather than the user's or the image's. Unfortunately, on-line help systems and manuals (assuming that users can be relied upon to read them) do not solve the vocabulary problem. Terms used in help systems and manuals can be just as cryptic as labels in interfaces; for instance,

in versions of Microsoft Excel, novices can't use the on-line help system to discover how to create a graph unless they are able to figure out that graphs are called *charts*.

The mapping of input to meaning is even more complicated for spoken dialogue and NL systems than it is for command languages. This is because even when the highest-frequency words are used, a simple message can still be expressed in an indefinite number of ways. To illustrate this idea: A group of computational linguists who were developing a natural language interface to a database query application tried to list all possible variations of a query that asked for *the set of programmers working for department managers*, using common words and syntax. Before they abandoned this enterprise, they managed to list no fewer than 7000 well-formed queries, all of which would seem perfectly natural in some discourse setting (the Hewlett-Packard Natural Language Project 1986, cited in Brennan 1990a). Figure 2 contains a representative sample from this list.

<<Figure 2>>

In short, the vocabulary problem seems to have no simple solution, and this seems especially true for spoken dialogue systems. Relying on a single, fixed mapping of words to referents works only in tightly constrained situations.[2] Relying on mappings that use the most “common” labels is not guaranteed to work either; the very idea of "word frequency" is deceptively simple. Estimates of word frequency are typically made with no attention to context of use, and yet language conventions are highly dependent on context of use. Despite popular assumptions by psycholinguists and computational linguists, there is no reliable default or null context. Context in dialogue is highly idiosyncratic and emerges from the common ground that accrues between two interactive partners.

Before turning to experiments about how people solve the vocabulary problem in conversation, I will describe some other manifestations of the vocabulary problem in message generation and in dialogue. I will also argue that the vocabulary problem is not unique to language-based interfaces.

2.2. The Vocabulary Problem in Direct Manipulation Interfaces

Direct manipulation interfaces are widely acknowledged to be easy to use for novices and occasional users. At the same time, command and natural language interfaces (whether spoken or textual) have been criticized for being underdetermined; that is, they can burden users with having to remember the precise set of terms and syntax that a system will accept (see, for instance, Shneiderman's (1992) critique of command and NL interfaces). Shneiderman has proposed that direct manipulation interfaces are superior to language-based interfaces, saying: "Since computers can display information 1,000 times faster than people can enter commands, it seems advantageous to use the computer to display large amounts of information and allow novice and intermittent users simply to choose among the items" (Shneiderman 1986). But novices and occasional users react badly to the clutter that a direct manipulation interface can spawn. One novice commented to me about the

profusion of icons in Microsoft Word 6.0: "The interface looks like the cockpit of an airplane. I miss my typewriter."

Furthermore, the vocabulary problem is by no means unique to language-based interfaces; it arises in direct manipulation because icons can be just as puzzling as labels. Even slight experience with an application makes people forget how ambiguous the mapping of symbol to function seemed in the very beginning. While observing undergraduate students' first-time encounters with popular applications, I had them report errors that occurred when they inferred the wrong mappings for icons. For instance, several students using Adobe Illustrator for the first time had no trouble using the magnifying glass icon to zoom in, but they all had trouble figuring out how to zoom out. One reasoned that the icon that resembled a "colored-in magnifying glass" should do the trick, but this seemed to do nothing at all. Another discovered a solution that was frustratingly inconsistent with zooming in: "You have to keep selecting the menu bar each time you zoom out, so this is much harder than zooming in." Another student was surprised to find that what looked like an eraser icon seemed to draw thin lines, while the pencil icon didn't seem to do anything. Another wondered why a fountain pen icon should draw spline curves, while a paintbrush icon should draw freehand lines. She also thought that an icon resembling scissors should be useful in cropping an image, but when she tried to use it that way, she got the following error message: "Can't scissor a path. Please use the scissors tool on a segment or an anchor point (but not an endpoint) of the path." The point is that pictures of tools are not inherently more "natural" cues than verbal commands; to appropriately map icons such as these onto functions requires experience with the application and its underlying metaphors.

2.3. The Vocabulary Problem in the Generation of Messages

Although Furnas et al. considered the vocabulary problem only in light of users' input commands to text editors, the problem also arises with output messages, as well as with the inferences users make relating outputs to inputs in a dialogue.

Error messages that use jargon. The vocabulary problem is guaranteed to arise when novices are unfamiliar with how words are used in a particular domain. For example, one student reported a white-knuckle experience while running Norton Disk Doctor, because she had no idea what the feedback message *Checking for Bad Blocks* meant. Her persistent search for a definition in the *Glossary* and *Topics* sections of the on-line help system turned up nothing; the term *block* wasn't even used in sections titled *Examining a Disk* and *Repairing a Disk*. The student finally asked a more experienced user and received a clear explanation in just a few sentences. Anecdotes like this one suggest an obvious but often missed opportunity for addressing the vocabulary problem in the form of on-line help: *An application should provide links to definitions of all the terms it uses*. Such definitions should be brief and use the same wording as the actual menu options and feedback terms presented by the application. Ideally, they should be sensitive to the current context the user finds herself in. This opportunity exists for direct manipulation interfaces

as well; one of the features of Adobe Illustrator that students found helpful was the text definition that appeared in the lower left when the mouse cursor was moved over an icon. A similar feature is the Macintosh's *Balloon Help*.

Lack of Internal Consistency. The vocabulary problem is exacerbated when an application has a lack of internal consistency. Not only is it important for an application to present a consistent model, but the labels applied to its parts and functions should be consistent throughout, including menu items, feedback messages, on-line help, paper manuals, and other documentation.

Lack of Parallelism in Dialogue. The vocabulary problem manifests itself in dialogue as well. In human conversation, the parts of common *adjacency pairs* such as questions and answers tend to parallel one another in their words and syntax (Levelt and Kelter 1982). Elliptical answers are common, as in this question/answer pair:

> Who has a terminal?

DAN FLICKINGER

Elliptical answers work well only when they are temporally or spatially adjacent to their questions. But adjacency is sometimes interrupted in human-computer dialogue. Here is the same query and a response from a prototype system that was designed to preserve parallelism as well as the context shared by questions and answers (Brennan 1988):

> Who has a terminal?

DAN FLICKINGER HAS A TERMINAL.

In contrast, consider this sample dialogue from a commercially available database query program, Symantec's QandA (Hendrix 1985):

> Who has a terminal?

SHALL I DO THE FOLLOWING? CREATE A REPORT SHOWING THE FULL NAME AND THE MANAGER AND THE EQUIPMENT FROM THE FORMS ON WHICH THE EQUIPMENT INCLUDES “TERMINAL"?

An answer that uses different words than its projecting question may lead an addressee to infer that a distinction is being made, or else that the speaker has misunderstood the question. In human-computer dialogue, even when the system interprets a query as a user has intended, the user may still have trouble understanding the system's response if it takes a different surface form than the query. This is another manifestation of the vocabulary problem.

Another version of the problem may arise because users tend to model their utterances on system messages. One of my students reported having the following exchange with Socrates, a command interface to a library database:

PLEASE SELECT COMMAND MODE

> Please find an author named Octavia Butler.

INVALID FOLIO COMMAND: PLEASE

Although the student was able to repair the problem, it arose in the first place because she expected the system to be able to understand the words that it *itself* had used. This is a reasonable expectation of a conversational partner (unless the partner is reading from a Berlitz phrasebook!), but one that is frequently violated in human-computer dialogue. This kind of error arises less often in direct manipulation interfaces, because their architectures support and require continuous representation of the objects of interest (Hutchins et al. 1986). This means that users can operate on the system's output representations and use them as input back to the systems (Draper 1986, has called this inter-referential I/O). But in the absence of an architecture that enforces parallelism between inputs and outputs, designers of spoken dialogue systems and other language-based interfaces manage this problem in more ad hoc way (when they manage it at all). This suggests another heuristic for coping with the vocabulary problem: *Applications should present as output only those terms and syntactic constructions they can process as input.*

Studies of spontaneous speech to human and computer partners show that it is a mistake to treat each utterance in the dialogue as independent and *ahistorical*, that is, produced and interpreted in the absence of an ongoing dialogue context. The next two sections examine the process of lexical choice in human-human and human-computer dialogues.

3. How People Solve the Vocabulary Problem in Conversation

According to the Principle of Contrast (E. Clark 1987), there is no such thing as a true synonym; even if two words seem interchangeable in a particular context, there are other contexts in which they contrast. Two terms applied to the same object often encode very different perspectives (see the expressions in Figure 1). When one speaker chooses a different term than another speaker in the same conversation, this can represent a kind of *implicit* correction, as in this spontaneous example reported by Jefferson (1982):

Ken: well- if you're gonna race, the police have said this to us

Roger: that makes it even better, the challenge of running from the cops!

Ken: the cops say if you wanna race, uh go out at four or five in the morning on the freeway...

A correction can be *explicit* as well, as in this example (Jefferson 1982):

Ken: hey the first time they stopped me from selling cigarettes was this morning.

Louise: from selling cigarettes?

Ken: or *buying* cigarettes.

When people persist in using different terms in the same conversation, this can mark a particularly salient contrast. This happens in combative contexts such as the courtroom, where participants may make a point of refusing to accept one another's perspectives. For instance, throughout the famous trial of a Boston physician who was charged with murder for performing an abortion, the prosecutor spoke of "the baby" while the defense lawyer spoke of "the fetus" (Danet 1980).

Such a contrast is noticeable because typically in a conversation, when two people discuss the same object repeatedly, they come to use the same terms. This phenomenon has been called *lexical entrainment* (Bortfeld and Brennan 1997; Brennan and Clark 1996; Garrod and Anderson 1987), and it has important implications for the vocabulary problem: While variability is high *between* conversations, it is relatively low *within* a conversation. Next I will back up these anecdotal observations with data from three experiments that systematically examine the process of lexical entrainment.

3.1. Lexical Entrainment in Human Conversation: Three Experiments

In our laboratory, we systematically explored word choice in a series of experiments (reported in detail in Brennan and Clark 1996). Pairs of people (a director and a matcher) who could not see each other were given the task of positioning the matcher's set of picture cards in the same order as the director's duplicate set. There were four picture cards of interest (a particular shoe, dog, car, and fish), and for these we examined the sequences of referring expressions people generated. In the first set of trials (A trials), the target cards were all unique in their basic level categories. In the second set of trials (B or non-unique trials), the card sets included the same four targets, but also additional shoes, dogs, cars, and fish. In the third set of trials (C trials), the card sets were the same cards as in the A trials. Because objects in the first C trial were unique in their basic level categories, traditional, *ahistorical* theories of referring would predict that at this point speakers should use unadorned basic level terms (*shoe*, *dog*, *car*, and *fish*), since these are informative enough and easy to produce (see Cruse 1977). In contrast, *historical* models of referring would predict that speakers' choices of expressions should depend on expressions used previously in the conversation.

In the first two experiments [3], people used the same terms in the first C trial as in the last B trial 46-52% of the time. This represents a substantial degree of lexical entrainment, considering that the last B and first C reference were separated, on average, by 11 other references (including some very long references to distracter items such as the one in Figure 1). When people did not use precisely the same terms in the first C trial, the terms they did use were usually strongly related to B terms (our coding criteria were strict, counting "the big red dog with the tongue hanging out" as different from "the big red dog with the tongue"). Often the first C expression was a shortened version of a B expression. This sequence illustrates shortening over five trials in one dialogue:

Trial B1 "a car, sort of silvery purple colored"

Trial B2 "purplish car going to the left"

Trial B3 "purplish car going left"

Trial B4 "the purplish car"

Trial C1 "the purple car"

Such shortening of expressions, as two people converge on a shared perspective and discussion becomes increasingly efficient, has been found in other studies that looked at referring within a static referent array (Clark and Wilkes-Gibbs 1986; Isaacs and Clark 1987; Krauss and Weinheimer 1964 1966; Schober and Clark 1989); it is a systematic feature of referring. It also happens in conversations between native and non-native speakers of English (Bortfeld and Brennan 1997). Shortening may happen gradually, as in the purple car example, or all in one step (see Carroll 1980). When speakers in our experiments did revert to unadorned basic level terms during the C trials, it was usually when their B terms had contained basic level terms within longer descriptions (as with "the man's brown shoe") rather than subordinate lexicalized terms (as with "the man's brown loafer"), and so using a basic level term in a C trial could be due to shortening a B term.

The fact that speakers often continue to use overinformative (as well as less common and sometimes much longer) terms identical or similar to the ones they have used previously within a conversation provides evidence against ahistorical models of referring that consider references to be independent events. The first half of Grice's maxim of quantity (being informative enough) is important, but the discourse history can outweigh the second half (being no more informative than necessary) when it comes to word choices.[4]

Next we considered which kind of historical explanation would best account for the lexical entrainment we observed. One proposal, the *output/input coordination principle* (Garrod and Anderson 1986), is strictly local; it predicts that a speaker formulates an utterance according to the same model and semantic rules used to formulate or interpret the most recent utterance. Another, more global, possibility

takes into account frequency of use by individual speakers; the more often people in conversation appeal to a particular mapping of word to concept, the more durable its representation in memory should be and the more likely it is to be used again. A third possibility includes frequency of use but makes a stronger claim: Lexical entrainment is specific to a pair of conversational partners. That is, speakers form conceptual pacts with particular addressees, which they mark by using consistent terminology.

In our experiments, the likelihood of lexical entrainment from the last B trial to the first C trial was strongly affected by total number of B trials (either one or four) that a pair had done. The more firmly a conceptualization had been established in the B trials, the more likely people were to appeal to it in the C trials. If they had simply been perseverating on their most recent terms, then the number of previous B trials should not have mattered. So a simple recency account such as Garrod and Anderson's output/input coordination principle does not explain our results.

To distinguish the other two possibilities, we ran a third experiment in which directors and matchers did no A trials, four B trials (with the non-unique cards), and then four C trials (with the unique cards, just as in the first two experiments). But half of the directors received a brand-new partner (who had never seen the cards before) for the C trials, while the other half continued with the same partner. If people were really forming conceptual pacts with specific partners, there should be more overinformative terms in the C trials when they continued to match cards with the same partner than with a new partner.

The results showed such partner-specific effects; directors with continuing matchers were more likely to appeal to conceptual pacts from the B trials than did directors with new matchers. And directors with new matchers were more likely to switch to unadorned basic level terms in the C trials. In the conversational transcripts, we found abundant evidence for referring as a collaborative process (see Clark and Wilkes-Gibbs 1986). Matchers (M) influenced the terms that directors (D) used, as in this example:

D: a docksider

M: a what?

D: u m

M: is that a kind of dog?

D: no, it’s a kind of um leather shoe, kinda preppy pennyloafer

M: okay, okay, got it

After this exchange, the director continued to refer to this object as "the pennyloafer." Even though the director was the one who knew which objects

needed to go where, sometimes the conceptualizations they ended up agreeing upon were proposed by matchers, as in this example:

D: another fish, the most realistic looking one with the pink stripes, green and pink

M: a rainbow trout?

D: yeah, yeah

Examples like these support the idea of conceptual pacts in referring. When a speaker chooses a referring expression, she is proposing a conceptualization that her addressee may or may not agree to. So a referring expression is provisional until it is ratified or else modified by a partner. Hedges seem to be one way of marking that a referring expression is provisional. In our first two experiments, we found that directors were more likely to use hedges in B trials, when many alternative conceptualizations were possible, than in A and C trials, when the possibilities were limited by objects' uniqueness in their basic level categories or by lexical pacts.

Although we found conceptual pacts to be partner-specific, we do not claim that speakers maintain distinct vocabularies that they download when they re-enter a conversation with a particular partner (although it is possible that the effect is due *in part* to the partner acting as a memory cue for a previously established pact). It is likely that the addressee's expectations play a major role in shaping the terms a speaker chooses. In our third experiment, the new matchers in the C trials had not participated in or witnessed the pacts from the B trials, and so probably expected to hear basic level terms. Consider this example where a director started to use an overinformative expression with a new matcher (the pair of asterisks denotes overlapping speech):

D: it's a fish with *different colors*

M: *yeah* okay

By interrupting the director with yeah, the matcher let him know that the basic level term was good enough. Thereafter, this pair used *fish*. Examples like these show that term-to-referent mappings need not result from the planning processes of one speaker, but can emerge from the conceptual coordination of two people.

In sum, conceptual pacts are flexible and temporary agreements to conceptualize an object in a particular way. That referring involves forming conceptual pacts is consistent with a model of conversation presented by Clark and Schaefer (1987, 1989), in which contributions to conversation are structured into two phases: a presentation phase and an acceptance phase. Once people have established a conceptual pact, they mark it by reusing the same or strongly related terms. They continue to appeal to it in later references even when simpler references are possible, and they are more likely to do this the more firmly the pact has been

established. A conceptual pact does not come about simply because certain terms are primed in a speaker's memory; it is established and maintained through interacting with a particular addressee.

3.2. What Conversational Data Predict about Spoken Dialogue Systems

It is clear from studies of human conversation that people have considerable resources for solving the vocabulary problem when they are talking to other people. Whether they are referring to ambiguous figures such as the one in Figure 1, or common objects such as shoes, dogs, cars, and fish, their referring expressions emerge from an interactive process in which one conversational partner makes a proposal that is then modified or ratified by the other. On this view, meanings do not represent fixed, determinate, or default mappings of words to models. The mappings are negotiable, so that meanings are achieved by speakers and addressees acting jointly.

This view is further supported by some surprising data about overhearers. The act of participating in a conversation (as opposed to passively overhearing it) has measurable effects on comprehension of referring expressions. In a study by Schober and Clark (1989) using the same kind of conversational task and abstract geometric objects we used in Figure 1, matchers who overheard but did not participate in conversations between directors and other matchers performed more poorly than did the matchers who were allowed to speak. This happened because sometimes the overhearers would understand which target card the director was referring to before the participating matcher did, but at other times, they would fall behind the other matcher, and would not be able to clarify which object the director meant. This result underscores the fact that successful communication is not simply a transfer of messages, but requires conceptual coordination. Such coordination is achieved by an interactive process we have described elsewhere, called *grounding* (Clark and Brennan 1991; Clark and Wilkes-Gibbs 1986; Clark and Schaefer 1987, 1989; Schober and Clark 1989). During the grounding process, two people in conversation seek and provide evidence about how utterances have been understood (Brennan 1990b).

Can results from these referential communication studies be applied to spoken dialogue systems? Perhaps human conversations and human-computer dialogues are too different to be compared, and people will avoid transferring their expectations from conversation to their interactions with computers (as Guindon and her colleagues suggested). The evidence I have presented so far argues against this. While people do not transfer all of their expectations, they do transfer some.

For example, the Wizard of Oz study I mentioned earlier that found just as many third-person pronouns addressed to computers as to other people (Brennan 1991) also showed that people expect topical and anaphoric continuity in dialogues with computers, just as with human partners. At the same time, this study found almost no first- or second-person pronouns in human-computer dialogues, compared to an abundance in human conversations. Queries that use such pronouns (e.g., "I need to know. . . ." or "Can you tell me. . . .") acknowledge a social context, which was

apparently absent with computer partners. While the wizard in our study was blind to whether the subjects believed she was a person or a computer and greeted them all in exactly the same way, the subjects began the conversations differently - they always greeted human partners, while they greeted computer partners only half the time. They almost always started out by directing complete-sentence, grammatical queries to human partners, and abbreviated, telegraphic strings to computer partners. Interestingly enough, this changed over the course of the conversations. In this study, the simulated computer and human partners responded with either complete sentence answers or with shortened, elliptical ones. By the last half of the session, people tended to adopt the same syntactic styles as their partners (either grammatical or shortened). This is a kind of *syntactic* entrainment. It is probably mediated by both automatic and strategic cognitive processes. Syntactic priming has been demonstrated in production experiments, in which speakers who describe pictures are more likely to produce sentences in either active or passive forms after being primed with the same forms (Bock 1986). So this could be an automatic cognitive process that has nothing to do with discourse. On the other hand, it could also be a strategic discourse process in which people avoid unwanted implicatures and let their partners know that they believe they are both talking about the same thing.

So it seems reasonable to ask to what degree people's behavior and expectations in discourse with human partners are similar to their behavior and expectations in discourse with computer partners. Lexical variability can be just as high across different human conversations as across different human-computer contexts. For instance, in our second experiment, the likelihood that two particular speakers in different conversations would choose the same terms in the B (non-unique) trials was only 10% (Brennan and Clark 1996); this is in the same range that Furnas et al. (1987) found for command languages. Because automatic, autonomous processes such as priming must play some role in lexical entrainment, there should be some similarities. But because people are likely to conceptualize a computer partner (and the act of talking with one) differently than a human partner, there should also be some differences.

In the next section, I will describe two more studies that used a Wizard of Oz paradigm to examine text and spoken language dialogues with computers (Brennan et al. 1997). We set out to discover whether there would be less lexical variability *within* than *between* dialogues, due to lexical convergence with a system. Note that I use the term "lexical convergence" rather than "lexical entrainment," to allow for the possibility that adopting a person's term may result from a different process than adopting a system's term; most systems are not in any position to negotiate, and most users are probably aware of this.

4. Lexical Convergence in Dialogues with Computers

We conducted two Wizard of Oz experiments using a database query task. The first simulated a text-based natural language interface and the second, a speech

recognition interface (with synthesized speech output). Users asked questions about the missing information in a small database of fictitious people and their attributes (such as what cars they drove, what kinds of homes they lived in, where they had gone to college, etc.). The database was represented by a spreadsheet with some of its values missing. Columns were labeled with names; rows, which were unlabeled, represented attributes. Attributes were unlabeled because we wanted users to infer them from the values present in the spreadsheet and then generate their own terms when they first referred to them.

The first variable we manipulated was the system's response style. When a user asked a question such as "What college does Aida attend?" she got one of three kinds of responses:

(1) "The college Aida attends is Williams." *(response that parallels query)*

(2) "The school Aida attends is Williams." *(implicit correction)* or

(3) "By college, do you mean school? *(explicit correction)*

After the user responded to the helpful error message in (3) with "yes," the system then responded as in (2). Each user experienced all three kinds of responses. For responses (2) and (3), the experimenter/wizard simply used a different term than the one first presented by the user. These response strategies were inspired by the examples cited earlier of implicit and explicit correction in Jefferson's (1982) conversation analysis study. In both our text and speech experiments, the system's response was parallel to the user's query 1/3 of the time, an implicit correction 1/3 of the time, and an explicit correction 1/3 of the time. After a user's first reference to an attribute, she had three more chances to refer to it. We expected that people would be more likely to adopt the system's term after an explicit correction than after an implicit one. We were curious as to whether they would adopt the system's term after an implicit correction, since the system had "understood" their original term, and so they were under no obligation to change.

The second variable was the extent to which memory for the partner's term would play a role in lexical convergence. We manipulated whether the opportunity for re-referring took place either immediately or after a delay in which the user referred to several other objects. Although all users received the same database spreadsheet, one group of users was instructed to retrieve the missing information by going horizontally across the spreadsheet from left to right (the immediate condition), while the other group was instructed to go down the spreadsheet vertically (the delay condition) and so asked several other questions before having the opportunity to re-refer to an attribute.

Results. Whenever the system had used a different term than the user's first term, we coded whether the user adopted the system's term in subsequent queries about the same attribute. The results I will describe represent what people did on their second reference to each database attribute (that is, their first *re*-reference). The

general pattern of results was similar for both text and speech interfaces (see Figures 3 and 4). As predicted, people were more likely to adopt the system's term at the first opportunity after an explicit than implicit correction (respectively, 94 to 37% for text and 88 to 58% for speech). That they adopted the system's terms as often as they did after implicit corrections is interesting, since the system interpreted their original terms without error messages, and so they were under no obligation to give up their own terms. To the extent that they adopted the system's terms after these implicit corrections, lexical convergence may be automatic.[5]

<<Figures 3 and 4>>

The immediate and delayed memory conditions affected lexical convergence as well; people were more likely to adopt the system's term when they re-referred to the same item immediately than when they did so after a delay of several other questions (72 to 59% for text and 87 to 59% for speech). While the direction of these differences is the same for text and speech, visual comparison of Figures 3 and 4 shows that when the implicit correction condition is considered alone, immediacy made twice as much of a difference for speech as for text (38 to 19%). I compare the text and speech results with some caution, since they come from two different experiments; however, the database query tasks were nearly identical and the participants in both were sampled from the same population of naïve undergraduates at the State University of New York at Stony Brook.

The implicit correction/delayed memory condition for speech is the one most comparable to the human conversation experiments described earlier (and in Brennan and Clark 1996), and it is worth noting that the likelihood of lexical convergence with a computer partner in this condition (58%) is close to the 46-52% likelihood of lexical entrainment with a human partner. People appear to be *at least as likely* to adopt the terms of computer partners as they are to entrain on terms with human partners. In another, text-only Wizard of Oz study in our laboratory with a slightly different task, in which lexical convergence with computer partners was directly compared to that with human partners, people actually adopted the computer's terms more often than they adopted other people's terms (Ohaeri 1995). Ohaeri (1995) has proposed that when people adopted the system's terms, they did so out of different kinds of strategies than the ones they used with human partners, such as to avoid future errors (because they expected the system to be inflexible).

Just as users model the syntactic form or length of their input utterances on the system's output messages (Brennan 1991; Brennan and Ohaeri 1994; Dybkjær et al. 1995; Zoltan-Ford 1991) and on the task scenarios they are given (Dybkjær et al. 1995), they also choose their words based on the system's messages. In our studies, lexical convergence occurred in text dialogue systems, but was particularly strong in spoken dialogue systems. It occurred a whopping 87% of the time when people had an immediate opportunity to adopt the system's terms.

5. Solving the Vocabulary Problem: Implications for Spoken Dialogue Systems

In conclusion, the bad news is that the vocabulary problem is ubiquitous–in conversation, in human-computer dialogue of all sorts, and especially, in spoken dialogue systems. The good news is that just as in human conversation, there is far less variability within dialogues than between, due to the likelihood of lexical convergence. This finding suggests that historical dialogue models can be useful in predicting and constraining the words and syntactic constructions people choose when talking to spoken dialogue systems, as well as in providing guidance for word choices in system messages. What follows is a summary of strategies for designers of spoken dialogue systems to consider in managing the vocabulary problem. Some of these apply to solving the vocabulary problem in human-computer interaction more generally; others are specific to spoken dialogue systems.

If the terminology an application uses is fixed in advance by the system designer, then it should be used consistently throughout the application. References to the same objects should use the same terms. Consistency in referring should be maintained not only in the system's spoken messages, but also in any text aspects of the interface, as well as in on-line or printed help or documentation about the system.

Prompts and output messages should use only those terms and syntactic constructions that the spoken dialogue system can process as input. This provides a natural and painless way to constrain the input and avoid errors, since people often model their lexical and syntactic choices on the system's utterances. This heuristic is especially important for those systems that do not explicitly prompt people with the words that can be used as input.

Applications should provide links to definitions of terms. Whenever possible, such definitions should be context sensitive and they should make the mappings of words to concepts clear. Where appropriate, the grammar of a spoken dialogue system could allow for ways to ask for expanded definitions of terms used in system messages (such as *what is <term>* or *what do you mean by <term>*, etc.) and users could be informed about this feature.

When appropriate, the system should take the initiative and provide explicit choices. One approach to the vocabulary problem that works in some situations, and also addresses the common criticism that speech interfaces are underconstrained, is to have the system take as much of the initiative in a dialogue as possible. When necessary, input utterances can be constrained explicitly. Even when a spoken dialogue system asks a yes-no question (which may constrain responses only implicitly), there is no guarantee that users will respond with *yes* or *no*; in one study of people receiving collect calls (reported by Kamm 1993), only 55% responded with a *yes* or *no* to the query, "Will you accept the call?" When a menu-like prompt was added ("Say *yes* if you will accept the call, otherwise say *no*"), more than 80%

responded with *yes* or *no*. Telephone interfaces that enable the general public to use speech to accept collect telephone calls use this strategy.

Spoken language systems should be supported by an architecture that explicitly supports (and expects) feedback and negotiation. Because of the ubiquity of the vocabulary problem, feedback and negotiation are necessary in any spoken dialogue system that aims to handle spontaneous utterances from real users. Actions ordinarily classified as "errors" or "repairs" are actually a necessary and natural part of robust communication. What is known about the process of grounding in human conversation can be applied to feedback in dialogue systems, such as with the adaptive feedback model proposed in Brennan and Hulteen (1995) or in an implemented system such as the MailCall Messaging System (Marx, this volume). The more the architecture of a spoken dialogue system supports the grounding process, the more likely that interacting with it will be as smooth and robust as interacting with a well-executed direct manipulation system (Brennan 1997).

Finally, the results presented here suggest an agenda for further research. Spoken dialogue systems that use strategies based on lexical entrainment and lexical convergence for managing vocabularies could be prototyped and tested. For instance, a system could begin a dialogue with a large (and inefficient) vocabulary, in order to allow a user to propose terms. Negotiation–either implicit or explicit– of the mappings between referring expressions and meanings would be necessary early in the dialogue; while this might take considerable effort, particularly with a novice user, it would enable more efficient referring later on. The system would need to be able to ground terms with the user, to ensure that both of them were reasonably confident they were mapping terms onto compatible conceptualizations (analogous to how people form conceptual pacts in conversations with other people). As the dialogue proceeded, the system would maintain a discourse model of currently active conceptualizations, terms, and mappings to rapidly narrow down the vocabulary its speech recognizer expected, as well as to constrain its word choices in generation. The discourse model should also allow for the shortening of expressions, which happens frequently upon re-referring. The system should remember the conceptualizations, terms, and mappings that it used with a particular user, and then start a new dialogue with that user by re-evoking the same model. Of course, such strategies brings with them other problems, such as how to re-expand the vocabulary when the topic under discussion changes (just as in human conversation), but I believe that they are worth a try. Until spoken dialogue systems are able to treat dialogues as historical and to perform coherently as dialogue partners, they will not reach their full potential in the human-computer interface.

Acknowledgments

I thank my collaborators: Heather Bortfeld, Herbert Clark, Eric Hulteen, Greg Lee, Justina Ohaeri, Pamela Stellmann Reis, Claire Rubman, and Michael Schober. This material is based upon work supported by the National Science Foundation under

Grants No. IRI9202458 and IRI9402167 and by Apple Computer, Inc. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation or Apple Computer, Inc.

References

Bock, J. K. (1986). Syntactic persistence in language production. *Cognitive Psychology, 18*, 355-387.

Bortfeld, H. & Brennan, S. E. (1997). Use and acquisition of idiomatic expressions in referring by native and non-native speakers. *Discourse Processes, 23,* 119-147.

Brennan, S. E. (1988). The multimedia articulation of answers in a natural language database query system. *Proceedings, Second Conference on Applied Natural Language Processing*, pp. 1-8. Association for Computational Linguistics, Austin, TX.

Brennan, S. E. (1990a). Conversation as direct manipulation: An iconoclastic view. In B. K. Laurel (Ed.), *The art of human-computer interface design*, Addison-Wesley, Reading, MA.

Brennan, S. E. (1990b). Seeking and providing evidence for mutual understanding. Unpublished Ph.D. dissertation, Stanford University, Stanford, CA.

Brennan, S. E. (1991). Conversation with and through computers. *User Modeling and User-Adapted Interaction*, *1*, 67-86.

Brennan, S. E. (1996). Lexical entrainment in spontaneous dialog. *Proceedings, ISSD 96, International Symposium on Spoken Dialog*. October 2-3, Philadelphia, PA, 41-44.

Brennan, S. E., & Clark, H. H. (1996). Lexical choice and conceptual pacts in conversation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *22*, 1482-1493.

Brennan, S. E. & Ohaeri, J. O. (1994). Effects of message style on users' attributions toward agents. *CHI '94, Human Factors in Computing Systems, Conference Companion*, 281-282.

Brennan, S. E., Ries, P. S., Rubman, C., & Lee, G. (1997). The vocabulary problem in speech and language interfaces. Unpublished manuscript.

Brown, R. (1958). *Words and things*. The Free Press, Glencoe, IL.

Carroll, John M. (1980). Naming and describing in social communication. *Language and Speech, 23*, 309-322.

Clark, E. V. (1987). The principle of contrast: A constraint on language acquisition. In B. MacWhinney (Ed.), *Mechanisms of language acquisition*. Hillsdale, NJ: Erlbaum.

Clark, H. H., & Brennan, S. E. (1991). Grounding in communication. In L. B. Resnick, J. Levine, and S. D. Behrend (Eds.), *Perspectives on socially shared cognition* (pp. 127-149). Washington, DC: APA Books.

Clark, H. H., & Schaefer, E. F. (1987). Collaborating on contributions to conversations. *Language and Cognitive Processes*, *2*, 19-41.

Clark, H. H., & Schaefer, E. F. (1989). Contributing to discourse. *Cognitive Science, 13*, 259-294.

Clark, H. H., & Wilkes-Gibbs, D. (1986). Referring as a collaborative process. *Cognition, 22*, 1-39.

Cruse, D. A. (1977). The pragmatics of lexical specificity. *Journal of Linguistics, 13,* 153-164.

Danet, B. (1980). 'Baby' or 'fetus'?: Language and the construction of reality in a manslaughter trial. *Semiotica, 32*, 187-219.

Draper, S. W. (1986). Display managers as the basis for user-machine communication. In D. A. Norman & S. W. Draper (Eds.), User Centered System Design, pp. 339-352. Hillsdale, NJ: Erlbaum.

Dybkjær, L., Bernsen, N. O., & Dybkjær, H. (1995). Scenario design for spoken language dialogue systems development. *Proceedings of the ESCA workshop on spoken dialogue systems*, Vigsø, Denmark, May 30-June 2, 1995, 93-96.

Deutsch, W. & Pechmann, T. (1982). Social interaction and the development of definite descriptions. *Cognition, 11*, 159-184.

Ford, W. & Olson, D. (1975). The elaboration of the noun phrase in children's descriptions of objects. *Journal of Experimental Child Psychology, 19*, 371-382.

Furnas, G. W., Landauer, T. K., Gomez, L. M., & Dumais, S. T. (1983). Statistical semantics: Analysis of the potential performance of keyword information systems. *Bell Systems Technical Journal*, *62*, 1753-1806.

Furnas, G. W., Landauer, T. K., Gomez, L. M., & Dumais, S. T. (1987). The vocabulary problem in human-system communications. *Communications of the ACM, 30*, 964-971.

Garrod, S. & Anderson, A. (1987). Saying what you mean in dialog: A study in conceptual and semantic co-ordination, *Cognition, 27*, 181-218.

Good, M. D., Whiteside, J. A., Wixon, D. R., & Jones, S. J. (1984). Building a user-derived interface. *Communications of the ACM, 27*, 1032-1043.

Grice, H. P. (1975). Logic and conversation (from the William James lectures, Harvard University, 1967). In P. Cole, & J. Morgan (Eds.), *Syntax and semantics 3: Speech acts* (pp. 41-58), Academic Press, New York.

Guindon, R. (1991). Users request help from advisory systems with simple and restricted language: Effects of real-time constraints and limited shared context. *Human-Computer Interaction*, *6*, 47-75.

Guindon, R., Shuldberg, K., & Conner, J. (1987). Grammatical and ungrammatical structures in user-adviser dialogues: Evidence for sufficiency of restricted language in natural language interfaces to advisory systems. *Proceedings, 25th Annual Meeting of the ACL*, pp. 41-44. Association for Computational Linguistics, Stanford, CA.

Hendrix, G. (1985). Q&A. Software, Symantec.

Hutchins, E. L., Hollan, J. D., and Norman, D. A. (1986). Direct manipulation interfaces. In D. A. Norman & S. W. Draper (Eds.), User Centered System Design, pp. 87-124. Hillsdale, NJ: Erlbaum.

Isaacs, E. & Clark, H. H. (1987). Reference in conversation between experts and novices. *Journal of Experimental Psychology: General*, 116, 26-37.

Jefferson, G. (1982). On exposed and embedded correction in conversation. *Studium Linguistik*, *14*, 58-68.

Kamm, C. (1993). User interfaces for voice applications. Colloquium presentation, *Human-machine communication by voice*, National Academy of Sciences, Irvine CA, Feb. 8-9.

Katzenberg, B. & Piela, P. (1993). Work language analysis and the naming problem. *Communications of the ACM, 36*, 86-92.

Krauss, R. M., & Weinheimer, S. (1964). Changes in reference phases as a function of frequency of usage in social interaction: A preliminary study. *Psychonomic Science*, *1*, 113-114.

Krauss, R. M., & Weinheimer, S. (1966). Concurrent feedback, confirmation, and the encoding of referents in verbal communication. *Journal of Personality and Social Psychology*, *4*, 343-346.

Levelt, W. J. M., & Kelter, S. (1982). Surface form and memory in question answering. Cognitive Psychology, 14, 78-106.

Makhoul, J. (1993). Overview of speech recognition technology. Colloquium presentation, *Human-machine communication by voice*, National Academy of Sciences, Irvine CA, Feb. 8-9.

Mangold, R. & Pobel, R. (1988). Informativeness and instrumentality in referential communication. *Journal of Language and Social Psychology, 7*, 181-191.

Ohaeri, J. O. (1995). Lexical convergence with human and computer Partners: Same cognitive process? Master's thesis, Department of Psychology, SUNY, Stony Brook, NY.

Olson, D. R. (1970). Language and thought: Aspects of a cognitive theory of semantics, *Psychological Review, 77*, 257-273.

Pechmann, T. (1989). Incremental speech production and referential overspecification. *Linguistics, 27*, 89-110.

Pinker, S. (1994). *The language instinct*. New York: William Morrow and Company, Inc.

Rosch, E., Mervis, C. B., Gray, W., Johnson, D., & Boyes-Braem, P. (1976). Basic objects in natural categories, *Cognitive Psychology*, *8*, 382-439.

Schmandt, C. M. & Marx. M. (in press). Speech recognition issues in spoken dialogue interfaces. In S. Luperfoy (Ed.), Automated Spoken Dialogue Systems. Cambridge, MA: MIT Press.

Shneiderman, B. (1986). The future of interactive systems and the emergence of direct manipulation. *Behavior and Information Technology, 1*, 237-56.

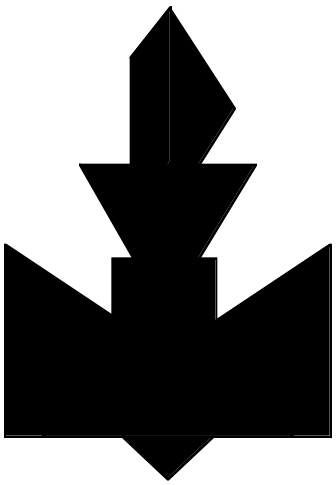
Shneiderman, B. (1992). *Designing the user interface* (second edition). Addison-Wesley, Reading, MA.

Schober, M. F., and Clark, H. H. (1989). Understanding by addressees and overhearers. *Cognitive Psychology, 21*, 211-232.

Stellmann, P. & Brennan, S. E. (1993). Flexible perspective-setting in conversation. *Abstracts, 34th Annual Meeting of the Psychonomic Society*, Washington, DC.

Zoltan-Ford, E. (1991). How to get people to say and type what computers can understand. *International Journal of Man-Machine Studies, 34*, 527-547.

Figure 1: Referring Expressions from 13 Different Conversations about the Same Figure (Stellmann & Brennan, 1993)



"a bat"
"the candle"
"the anchor"
"the rocket ship"
"the Olympic torch"
"the Canada symbol"
"the symmetrical one"
"shapes on top of shapes"
"the one with all the shapes"
"the bird diving straight down"
"the airplane flying straight down"
"the angel upside down with sleeves"
"the man jumping in the air with bell bottoms on"

Figure 2: Excerpt from 7000 Variations on a Single Sentence (Brennan, 1990a)

List programmers department managers supervise.
What programmers work for department managers?
List programmers working for department managers.
List programmers who work for department managers.
List any programmers department managers supervise.
List all programmers working for department managers.
List each programmer a department manager supervises.
Which programmers work for managers of departments?
Which programmers do department managers supervise?
List all programmers who work for department managers.
List all programmers that department managers supervise.
List programmers whose supervisors manage departments.
Which of the programmers work for department managers?
Who are the programmers department managers supervise?
List every programmer any department manager supervises.
List every programmer supervised by a department manager.
List programmers with supervisors who manage departments.
Which programmers are supervised by department managers?
Who are the programmers working for department managers?
List programmers whose supervisors are department managers.
List each programmer that any department manager supervises.
List all of the programmers who work for department managers.
Who are the programmers who work for department managers?
List every programmer whom a department manager supervises.
List each programmer who is working for a department manager.
Which programmers are there working for department managers?
Which of the programmers are department managers supervising?
Which of the programmers are working for department managers?
List each of the programmers supervised by a department manager.
List the programmers who are supervised by department managers.
Which of the programmers do managers of departments supervise?
Who are all of the programmers working for department managers?
Which of the programmers are supervised by department managers?
List any programmer whose supervisor is a manager of a department.
Who are the programmers being supervised by department managers?
Who are all of the programmers that department managers supervise?
List any programmers there might be working for department managers.
List everyone who is a programmer supervised by a department manager.
List each of the programmers who is supervised by a department manager.
Which of the programmers are being supervised by department managers?
List any programmer with a supervisor who is the manager of a department.
Who are the programmers whose supervisors are managers of departments?
Which of the programmers are being supervised by managers of departments?
List any programmer who has a supervisor who is the manager of a department.
List all programmers who work for anyone who is the manager of a department.
List all programmers working for supervisors who are managers of departments.
Which of the programmers have supervisors who are managers of departments?
List each of the programmers who is supervised by anyone managing a department.
Which of the programmers have supervisors who are the managers of departments?
Who are all of the programmers who have supervisors who are department managers?

Figure 3: Lexical convergence in text dialog with computers. *Delayed* and *Immediate* are the two memory conditions; *Implicit* and *Explicit* are the ways in which the system introduced a term that differed from the user's term.

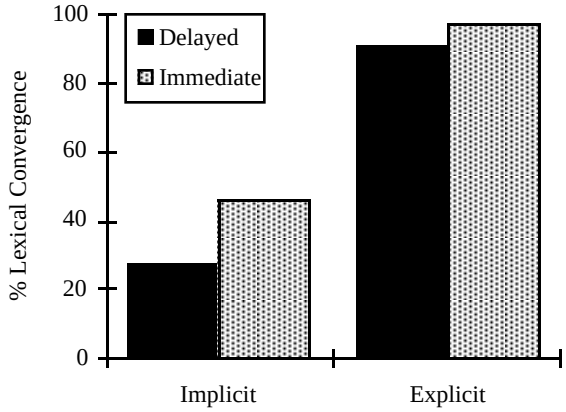


Figure 4: Lexical convergence in spoken dialog with computers. The memory conditions are the same as for Figure 3.