The Multimedia Articulation of Answers in a Natural Language Database Query System

Susan E. Brennan

Stanford University
and
Hewlett Packard Labs
1501 Page Mill Road
Palo Alto, CA 94304

Abstract

This paper describes a domain independent strategy for the multimedia articulation of answers elicited by a natural language interface to database query applications. Multimedia answers include videodisc images and heuristically-produced complete sentences in text or text-to-speech form. Deictic reference and feedback about the discourse are enabled. The interface thus presents the application as cooperative and conversational.

1 Introduction

It is useful to evaluate human-computer communication in light of Grice's cooperative principle and maxims [Gri75]. Recently there has been much interest in a "cooperative response" paradigm for interfaces to database query and expert systems [Ste87]. The most promising strategies in this area of investigation involve applying insights gained from psycholinguistics research in order to create better conversational human/computer interfaces. However, inventing adequate user modeling and inferencing systems for this purpose is no easy task, and much of the literature on the subject describes proposals for systems yet unimplemented or theoretical approaches which may depend heavily on a particular domain model. Our multimedia articulator consists of principled solutions which have been implemented in a domain independent manner and which produce answers that are reasonably relevant, informative, and conversational in style. Such a system makes it possible to begin to study users interacting with a question-answering application.

2 System overview

The system described here functions as a conversational human/computer interface to database query systems. It consists of a natural language front end and a module which articulates multimedia answers. The system accepts well-formed strings as input; these sentences are interpreted by an HPSG-based parser [PS87] which produces a parse tree. After further processing by a semantics module, a pragmatics processor [BFP87] and a disambiguator, a logical formula in the language NFLT [CP85] is produced. This formula is transduced into a database query. Two database query formats are currently supported: a frame-based representation language, HPRL, and the standard relational database query language, SQL. Answers returned from the database are then packaged appropriately by the articulator for presentation to the user.

The two database applications currently supported are a database of people and equipment (a subset of which we have proposed as a natural language evaluation test suite [FNSW87]), and a database of paintings by 19th century Dutch artist Vincent Van

Gogh and his contemporaries. The latter database was based on the index to a commercially available videodisc [Nim82] and augmented from other sources. Both applications can be run on workstations configured with or without multimedia output devices.

3 Database answer format

The driver of a database query application (i.e. the domain dependent part of the system) is responsible for returning answers in a list format which consists of a keyword specifying the type of the answer, followed by the answer itself. The answer types expected by the articulator are *boolean, number, item, set, quantity,* and *table.*

In deciding how to package a response, the articulator uses the answer type along with additional information provided by the parser which identifies the illocutionary act of a query as imperative, declarative, yes/no question, or wh-question. An answer is presented textually as a single phrase, as a complete sentence which parallels the user's query, or as a table. In addition, depending on answer type and the system's hardware configuration, an answer may include videodisc images, text-to-speech, icons and maps. While a user can request answers in a particular medium via menus, a default strategy is in place which yields a fairly satisfying style of human/computer interaction.

4 Text answers

4.1 Style

Questions and answers are a common kind of adjacency pair in human language use. The preferred style of an answer is often elliptical and shows parallelism with the surface syntactic structure of the preceding question [CC77]. In addition, lexical choice in the answer is constrained by that in the question. An answer which is articulated using different lexical entries than its projecting question may lead the user to infer that the system is making a distinction when it is in fact only using a synonym.

Although elliptical answers may be the norm in human/human conversation, the articulator described here defaults to "verbose mode"; it responds to most queries with complete sentence answers. The motivation for this approach arose when we noticed that shorter answers were unsatisfying in certain situations. When additional textual material intervenes on the user's screen after the input query is typed in and before the answer appears, and in other cases where the user is distracted or not watching the screen when the textual answer arrives, a short answer takes on something of the character of a non-sequitur. This problem manifested itself in an early version of our system that worked by having users send queries over the network via electronic mail to a single natural language server which in due time mailed its responses back to the user, and also in the current system, which returns most answers in a few seconds but can be operated in a mode which prints modular timing and status information during processing. Even more unsatisfying was the articulation of answers using text-to-speech hardware. Generated speech is often hard for users to understand [TRC84] and in our system, short answers delivered this way often failed even to attract a user's attention as information-bearing. To echo the query audibly seemed confusing; what was needed was the capability to frame the answer in a complete sentence based on the query. The final impetus for the verbose articulator was our desire to approximate some of the effects that real natural language generation capability might provide in a question-answering human/computer interface, before committing resources to a full-scale natural language generation effort.

In verbose mode, a sententi'al answer consists simply of a string derived from the formatted database answer with constituents of the user's original query wrapped around it. Articulation achieves the dual purposes of satisfying the user's request for information while preserving a conversational style of interaction (figure 1). It is interesting to compare these answers with the kind of paraphrasing capacity that one finds in some other systems which are commercially available (figure 2).

To paraphrase a user's query in a form that reflects the actual database access method (figure 1) can be extremely helpful in identifying misinterpretations of the query. However, that approach may interfere with

```
User:    Who has a terminal?

System:  DAN FLICKINGER HAS A TERMINAL.
```

Figure 1: Adjacency pair

```
User:    Who has a terminal?

System:  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"?"
```

Figure 2: Dialogue from Q&A [Hen85]

natural interaction by insisting that the user confirm his or her every conversational move. Furthermore, whether the system's interpretation of what the user meant by the query with respect to the database is a correct mapping or not, the user is forced to reformulate his or her question in a program-like or logical form. Such an interface imposes a significant cognitive load on the user. Presumably, a central motivation for providing a natural language interface to a database is to avoid forcing the user to use a foreign language. This strategy pays homage to Grice's maxim of *manner*, "avoid obscurity of expression". On the other hand, the argument has been made that separate, non-equivalent representations providing different views of the world should be maintained by the system [Spa83]; each of these views should be available to the user at appropriate times. Thus logical paraphrases, desirable in establishing initial system credibility, should be available upon specific request by the user.

4.2 "Namely" answers

Grice's maxim of *quantity* for cooperative communication is a reminder that it is frequently desirable to provide more information in an answer than was literally requested. For example, when a user asks "Are there any secretaries?" the best answer may be not "Yes", but "Yes - namely, X, Y, and Z" (where X, Y, and Z are the names of the secretaries). Several question-answering systems have addressed issues of this sort [WMJB83] [WJMM82]. While our system does not explicitly model the user's goals or know anything about indirect speech acts, it provides extended answers to some queries via a list containing the keyword *namely*, which appears as the last item in the answer list passed to the articulator (figure 3).

```
Sentence:     How many employees are there?
Answer list:  (NUMBER 4 (NAMELY {abrams}
              {chiang} {devito} {browne}))
Articulated:  THERE ARE 4 EMPLOYEES -
              NAMELY, IRA ABRAMS, LYN
              CHIANG, KAT DEVITO, AND
              DEREK BROWNE.
```

Figure 3: "Namely" answer

Extended answer lists are constructed as follows. When an answer is of type *number* and its cardinality is below a certain threshold, or else when it is both of type *boolean* and affirmative, the articulator makes an additional query to the database which returns information for constructing the "namely" answer. This additional information is combined with the short answer to the user's original query, to create an extended answer. In this way we attempt to comply with Grice's maxims of manner and quantity: to "be brief" and to "make your contribution as informative as is required".

4.3 Verbose mode

Verbose mode works as follows. Initially, a short answer string is created from the formatted list that the database returns. First, the type keyword is stripped off the answer list. Depending on the type, the remaining short answer list is transformed into a string which is a textual phrase consisting of one of the following: a name or names (for type *set* or *item* the database is queried and returns appropriate nouns or proper names), a string containing an integer (for type *number*), a string containing a number followed by units of measure (for type *quantity*), or the strings "yes" or "no" (for type *boolean*). *Set* answers are expanded into coordinated noun phrases with appropriate punctuation. If the type is *table*, a table is produced.

In constructing the short answers to wh-questions, some simple additional heuristics are used. First, if the short answer string was derived from a null set or null item, the answer is converted from the empty string to an appropriate string: "nowhere" if the wh-question word is "where", "never" for "when", "nobody's" for "whose", and either "none" or else "no" plus the string corresponding to the modified NP head for "which", "what" or "how many" phrases. Otherwise, when the answer is *not* an empty set and the wh-question word is "whose", " 's" is appended to the answer. When "whose" modifies the head of a noun phrase, the noun phrase is appended to the answer.

Then, once the short answer has been produced, if the query is not an imperative (and the answer is not a table), the input query's parse tree representation is transformed into a template with which to frame the short answer. Four functions traverse the parse tree and return strings corresponding to constituents from the input query: these constituents are *subject*, *auxiliary verb* (if there is one), *main verb phrase*, and *preposition* (if the wh-question word is within a prepositional phrase or fills a trace in one).

An *end-of-sentence* string is created which contains, simply, terminating punctuation, or else an expanded phrase consisting of "namely," followed by a coordinated noun phrase with appropriate punctuation. This expanded phrase is constructed whenever a short to medium-length *namely* list is available at the end of an answer list, as shown in figure 3.

Finally, the verbose answer string is constructed using one of two strategies: if the wh-question word is in subject position in the query, the constituents are positioned in the answer as follows, (the items in parentheses may or may not be present): *answer (aux-verb) (main-verb-phrase) (preposition) end-of-sentence*; if the wh-question word is in non-subject position, the positioning is: *subject (aux-verb) main-verb-phrase (preposition) answer end-of-sentence.*

If the query is a declarative or a yes/no question, a *boolean* answer results. When a boolean answer is affirmative, the string "yes," with the modified input string appended, is articulated. For negative boolean answers, if the input string contains an auxiliary verb, the following sequence is articulated: *"No," subject aux-verb "not" main-verb-phrase end-of-sentence* (figure 4). If there is no auxiliary verb in the negative answer, the canned phrase "No, it is not true that," with the original input string appended, is articulated. In addition, a some/any transformation is applied to yes/no questions. "Any of" is replaced by "none of" or "some of", depending on whether the answer is affirmative or negative. If the input query contains an auxiliary verb and the word "any" without "of", "any" is replaced by "no" or "some" (figure 5). If the constructed answer template contains successive double negatives (as might result from a query containing a negation), these are removed.

```
Sentence:      Does Ira program?
Answer list:   (BOOLEAN NIL)
Articulated:   NO, IRA DOES NOT PROGRAM.
```

Figure 4: Negation

```
Sentence:      Are there any consultants?
Answer list:   (BOOLEAN T (NAMELY {WASOW}
               {SAG}))
Articulated:   YES, THERE ARE SOME
               CONSULTANTS - NAMELY,
               TOM WASOW AND IVAN SAG.

Sentence:      Will any consultants work
               for Kat?
Answer list:   (BOOLEAN NIL)
Articulated:   NO, NO CONSULTANTS WILL WORK
               FOR KAT.
```

Figure 5: Some/any

Finally, contrast the situation where the answer list is (BOOLEAN NIL) with the one where the answer list is simply NIL (which means the database failed to return an answer). In this case, the system answers "I don't know whether" with the modified input query appended (figure 6).

The style of the articulator's verbose responses, while somewhat quaint, appears cooperative because the answer is delivered using the same lexical and syn-

```
Sentence:      Do any vice presidents work?
Answer list:   NIL
Articulated:   I DON'T KNOW WHETHER ANY
               VICE PRESIDENTS DO WORK.
```

Figure 6: Successful failure

```
Sentence:      Which manager is Kat Devito?
Answer list:   (ITEM)
Articulated:   KAT DEVITO IS NO MANAGER.
```

Figure 7: Pragmatic strangeness

```
Sentence:      WHAT SELF-PORTRAITS DID
               VAN GOGH PAINT?
Answer list:   (SET "F0296" "F0627" "F0522")
Articulated:   VAN GOGH DID PAINT
               SELF-PORTRAIT, SELF-PORTRAIT,
               AND SELF-PORTRAIT
               WITH GRAY FELT HAT.

Sentence:      SHOW ME STARRY NIGHT.
Answer list:   (ITEM "F0612")
Articulated:   STARRY NIGHT.
```

Figure 8: When words aren't enough

tactic forms that the user chooses in the query. Of course, this technique of wrapping the query around the answer works only in very simple question-answering applications, where the system has little of its own to say. Failure in the form of ungrammatical answers to wh-questions sometimes occurs due to lack of agreement; rather than extend the verbose articulator any further, it seems a better strategy to simply detect those cases and suppress an ungrammatical verbose answer in favor of a short one. Pragmatic failures that are still syntactically well-formed may also occur, particularly in negative boolean answers and empty set answers; we have not arrived at a consistently successful strategy for detecting and treating presuppositional failures (figure 7). Our implementation also does not take into account syntactic constraints on given/new information in framing the answer in the query. Despite these limitations, the appeal of verbose articulation argues for integrating a real generation capability with a natural language interface to database query.

5 Multimedia

While the articulator always manages to produce some sort of textual answer, it is often desirable to respond with an answer in a different medium (figure 8). Visual images from a videodisc can be displayed whenever item or set answers are associated with videodisc frames in the database, in addition to whenever an imperative is used to explicitly request images.

The articulator consults a module called *circus* which contains the drivers and methods pertaining to the videodisc player and the text-to-speech hardware. This module queries the database application to discover whether any entities in the answer list can be displayed as videodisc images. These images are represented and accessed by videodisc frame numbers which are stored in the database in SQL tables or in HPRL slots.

When the system is configured with the text-to-speech generator and the items in a set answer are associated with videodisc images, the entire textual answer is displayed first. Then a synchronizing function in *circus* articulates the items in the set by displaying the appoprate image on the video monitor and speaking the corresponding items, one at a time. Thus the user hears the name of an item spoken immediately after it comes up on the video monitor; videodisc images are displayed for a few seconds each. We have not synchronized the textual answers with the videodisc answers, since these media are displayed on two separate screens at somewhat different rates and it would be difficult for a user to attend simultaneously to both. Laser videodiscs in CAV format (constant angular velocity) advertise fast, random access to still images, yet with most videodisc players there is some time cost to searching for frames on a disc and for changing search direction. We minimize this cost by reordering the items in the set according to their videodisc frame numbers, which correspond to their ordering on the disc.

It seems appropriate to mention here that videodisc imagery, like sex and violence, can be either gratuitous or meaningful. In the course of our project, we have demonstrated both. In the context of our people and equipment database, the articulator is capable of displaying a picture of a featureless cubicle or a slide show of nervously posed employees in conjunction with a textual answer. On the other hand, the database of Van Gogh paintings has proven to be a very appealing application for visual articulation.

With visually articulated answers, we were provided with an opportunity to begin to experiment with deictic reference. While personal pronouns are interpreted by the pragmatics processor using a discourse model which takes a centering approach [Gro77] [Sid79] [JW81] [GJW83] [BFP87], demonstrative pronouns are interpreted via a rudimentary environment model that knows which painting is currently displayed on the video screen. Note that the displayed image may not be the one currently under discussion in the the discourse, but may be left over from an earlier query if no intervening queries elicited videodisc answers. Since imagery can be such a salient part of the user's environment, it is necessary to support deictic references to the current image. At present in our system, "this" and "that" have the same interpretation, but we are exploring alternatives such as interpreting "that" as referring to the *previously* displayed image when it appears contrastively in the same context as "this". A more thorough treatment should of course integrate spatial, temporal and discourse perspective [Lin79]. We are attempting to model more of the visual environment, including graphic elements on the screen, and to integrate deictic information more fully into the discourse.

By now it should be evident that one should not consider articulation of answers entirely independently from discourse. A natural language interface to a database query application can provide textual feedback about the discourse apart from the literal answer. Our articulator makes explicit the interpretation of the user's pronominal reference by substituting the phrase it cospecifies for the pronoun in the verbose answer (figure 9). Thus the user is likely to discover any misunderstanding instantly. On the other hand, since verbose answers rely on more or less blindly-applied heuristics to wrap text around the answer, the articulator is not a full partner in the discourse and is not capable of achieving subtle but nevertheless critical discourse functions through syntactic choices. A true generation component would presumably exercise lexical and syntax choices, thus avoiding eccentric as well as ungrammatical exchanges.

```
Q: What did Gauguin paint?
A: GAUGUIN PAINTED VINCENT PAINTING.
Q: How many pictures of Van Gogh were not
   painted by him?
A: 8 PICTURES OF VAN GOGH WERE NOT PAINTED
   BY GAUGUIN.
```

Figure 9: References made explicit

6 Conclusion

Obviously there is much ground to be covered in the areas of natural language communication and conversational human/computer interfaces. Yet interim applications can be built which are incrementally improved over previous ones. This approach is necessary in order to observe real users of these systems.

The domain independent articulation strategy presented here enables two very different database query systems to present answers conversationally. Generality is achieved through the use of answer type keywords (provided by the application driver) and the illocutionary act of the query (provided by the parser). From this information, multimedia answers are assembled and templates in which to frame the textual answer are constructed from the input query.

Although it lacks inferencing ability, the articulator described here provides several features desirable in a cooperative interface. These features include answers presented in a style that parallels the user's question, extended answers, the ability to refer deictically to an image, and explicit feedback regarding co-specifiers of personal pronouns.

Finally, multimedia articulation provides serendipitous opportunities for dispersing ambiguity, due to multiple representation of the answer. Take the following query to our Van Gogh database: "Show me the pictures of Van Gogh that he didn't paint." The textual answer came back: "The pictures of Van Gogh that Van Gogh didn't paint are Vincent Painting and Self-Portrait." As we puzzled over "Self-Portrait" (how could a self portrait be of Van Gogh, but not painted by him?) the videodisc answer was displayed on the adjacent screen: first, a portrait of Van Gogh that had been painted by his friend Gauguin, and – surprisingly – a self portrait of Van Gogh that was not *painted*, but *drawn*.

7 Acknowledgements

The work reported herein was jointly supported by the National Science Foundation and Hewlett Packard Labs. It was done in collaboration with HPLab's Natural Language group. I would especially like to thank Lew Creary, Dan Flickinger, Lyn Friedman and Herb Clark.

References

[BFP87] S.E. Brennan, M.W. Friedman, and C.J. Pollard. A centering approach to pronouns. In *Proc., 25st Annual Meeting of the ACL, Association of Computational Linguistics*, pages 155–162, Stanford, CA, 1987.

[CC77] H.H. Clark and E.V. Clark. *Psychology and Language.* Harcourt Brace Jovanovich, Publishers, 1977.

[CP85] L. Creary and C.J. Pollard. A computational semantics for natural language. In *Proc., 23st Annual Meeting of the ACL, Association of Computational Linguistics*, pages 172–179, Chicago, IL, 1985.

[FNSW87] D.P. Flickinger, J. Nerbonne, I. Sag, and T. Wasow. Toward evaluation of NLP systems (in conjunction with panel). In *25st Annual Meeting of the ACL, Association of Computational Linguistics*, Stanford, CA, 1987.

[GJW83] B.J. Grosz, A.K. Joshi, and S. Weinstein. Providing a unified account of definite noun phrases in discourse. In *Proc., 21st Annual Meeting of the ACL, Association of Computational Linguistics*, pages 44–50, Cambridge, MA, 1983.

[Gri75] H.P. Grice. Logic and conversation (from the William James lectures, Harvard University, 1967). In P. Cole and J. Morgan, editors, *Syntax and Semantics 3: Speech Acts*, pages 41–58, Academic Press, Inc., 1975.

[Gro77] Barbara.J. Grosz. *The representation and use of focus in dialogue understanding.* Technical Report 151, SRI International, 333 Ravenswood Ave, Menlo Park, Ca. 94025, 1977.

[GS85] B.J. Grosz and C.L. Sidner. *The structure of discourse structure.* Technical Report CSLI-85-39, Center for the Study of Language and Information, Stanford, CA, 1985.

[Hen85] G. Hendrix. Q&A. Software, Symantec, 1985.

[JW81] A.K. Joshi and S. Weinstein. Control of inference: role of some aspects of discourse structure - centering. In *Proc., International Joint Conference on Artificial Intelligence*, pages 385–387, Vancouver, B.C., 1981.

[Kap82] S.J. Kaplan. Cooperative responses from a portable natural language query system. *Artificial Intelligence*, 2(19), 1982.

[KJM86] J. Kalita, M. Jobes, and G. McCalla. Summarizing natural language database responses. *Computational Linguistics*, 12(2):107–124, 1986.

[Lin79] C. Linde. Focus of attention and the choice of pronouns in discourse. In T. Givon, editor, *Syntax and Semantics*, pages 337–354, Academic Press, Inc., 1979.

[LS85] W.G. Lehnert and S.P. Schwartz. Data base querying by computer. In A.C. Graesser and J.B. Black, editors, *The Psychology of Questions*, pages 359–374, Lawrence Erlbaum Associates, Publishers, 1985.

[Nim82] L. Nimoy. Vincent Van Gogh: a portrait in two parts. Videodisc, Philips International/North American Philips Corporation, 1982.

[Pol85] M. Pollack. Information sought and information provided. In *CHI '85*, pages 155–160, San Francisco, CA, 1985.

[PS87] C. Pollard and I.A. Sag. *Information-Based Syntax and Semantics. Vol. 1: Fundamentals.* (in press). Lecture notes series no. 13, Center for the Study of Language and Information, Stanford, CA, 1987.

[Rei85] R. Reichman. *Getting Computers to Talk Like You and Me.* MIT Press, Cambridge, MA, 1985.

[Sid79] Candace L. Sidner. *Toward a computational theory of definite anaphora comprehension in English.* Technical Report AI-TR-537, MIT, 1979.

[Sid81] C.L. Sidner. Focusing for interpretation of pronouns. *American Journal of Computational Linguistics*, 7(4):217–231, 1981.

[Spa83] K. Sparck-Jones. Shifting meaning representations. In *Proc., International Joint Conference on Artificial Intelligence*, pages 621–623, 1983.

[Ste87] P. Stenton. *Designing a co-operative interface to an expert system.* Technical Report HPL-BRC-TM-87-023, HPLabs Technical Memo, 1987.

[TRC84] J.C. Thomas, M.B. Rosson, and M. Chodorow. Human factors and synthetic speech. In *Proc., INTERACT '84*, pages 37–42, 1984.

[WJMM82] B. Webber, A. Joshi, E. Mays, and K. McKeown. Extended natural language data base interactions. *International Journal of Computers and Mathematics: Special issue on computational linguistics*, 1982.

[WMJB83] W. Walster, H. Marburger, A. Jameson, and S. Busemann. Over-answering yes-no questions: extended responses in a NL interface to a vision system. In *Proc., International Joint Conference on Artificial Intelligence*, pages 643–646, 1983.