

Is There Natural Language after Data Bases?

Jaime G. Carbonell
Computer Science Department
Carnegie-Mellon University
Pittsburgh, PA 15213

1. Why Not Data Base Query?

The undisputed favorite application for natural language interfaces has been data base query. Why? The reasons range from the relative simplicity of the task, including shallow semantic processing, to the potential real-world utility of the resultant system. Because of such reasons, the data base query task was an excellent paradigmatic problem for computational linguistics, and for the very same reasons it is now time for the field to abandon its protective cocoon and progress beyond this rather limiting task. But, one may ask, what task shall then become the new paradigmatic problem? Alas, such question presupposes that a single, universally acceptable, syntactically and semantically challenging task exists. I will argue that better progress can be made by diversification and focusing on different theoretically meaningful problems, with some research groups opting to investigate issues arising from the development of integrated multi-purpose systems.

2. But I Still Like Data Bases...

Well, then, have I got the natural language interface task for you! **Data base update** presents many unsolved problems not present in pure query systems. "Aha," the data base adherents¹ would say, "just a minor extension to our work!" Not at all; there is nothing minor about such an extension [4]. Consider, for example, the following update request to an employee-record data base:

"Smith should work with the marketing team and Jones with sales"

First, the internal ellipsis in the coordinate structure is typical of such requests, but is mostly absent from most DB queries. However, let us assume that such constructions present no insurmountable problems, so that we can address an equally fundamental issue: What action should the system take? Should Smith be deleted from sales and added to marketing (and vice versa for Jones)? Or, should Smith and Jones remain fixed points while all other sales and marketing employees are swapped? As Kaplan and Davidson [3] point out, one can postulate heuristics to ameliorate the problem. They proposed a minimal mutilation criterion, whereby the action entailing the smallest change to the

data base is preferred. However, their bag of tricks fails miserably when confronted with examples such as:

"The sales building should house the marketing people and vice versa"

Applying the above heuristic, the bewildered system will prefer to uproot the two buildings, swap them, and lay them on each other's foundations. Then, only two DB records need to be changed. Such absurdities can only be forestalled if a semantic model of the underlying domain is built and queried, one that models actions, including their preconditions and consequences, and knows about objects, relations, and entailments.

So, data base update presents many difficult issues not apparent in the simpler data base query problem. Why not, then, select this as the paradigmatic task? My only objection is to the definite article *the*-I advocate data base update as *one* of several theoretically significant tasks with major practical utility that should be selected. Other tasks highlight additional problems of an equally meaningful and difficult nature.

3. How Should I Select A Good Task Domain?

At the risk of offending a number of researchers in computational linguistics, I propose some selection criteria illustrated both by tasks that fail to meet them, and later by a much better set of tasks designed to satisfy these criteria for theoretical significance, and computational tractability.

1. The task should, if possible, be able to build upon past work, rather than addressing a completely disjoint set of problems. This quality enhances communication with other researchers, and enables a much shorter ramp-up period before meaningful results can be obtained. For instance, an automated poetry comprehension device fails to meet this criterion.
2. The task should be computationally tractable and grounded in an external validation test. Interfaces to as yet non-existent systems, or ones that must wait for radically new hardware (e.g., connectionist machines) before they can be implemented fail to meet this criterion. However, data base query interfaces met this criterion admirably.
3. The task should motivation investigation of a set of language phenomena of recognizable theoretical significance that can be addressed from a computational standpoint. Ideally, the task should focus on restricted instances of a general and difficult phenomenon to encourage progress towards initial solutions that may be extended to (or may suggest) solutions to the general problem. Data base query has been thoroughly

¹I must confess that I would have to include myself in any group claiming adherence to data base query as a unifying task. I am still actively working in the area, and to some extent expect to continue doing so. The practical applications are immense, but theoretical breakthroughs require fresh ideas and more challenging problems. Hence I advocate a switch based on scientific research criteria, rather than practical applicability or engineering significance.

mined for such phenomena; hence it is time to go prospecting on virgin land.

4. The task should be of practical import, or should be a major step towards a task of practical import. Aside from very real if mundane concerns of securing funding, one desires a large, eager, potential user community as an inexhaustible source of examples, needs, encouragement, and empirical motivation and validation. A parser for Summertime cuneiform tablets or a dialog engine built around the arbitrary rules of a talk-show game such as "You don't say" would completely fail on this criterion.

4. What Then Are Some Other Paradigmatic Tasks?

Armed with the four criteria above, let us examine some tasks that promise to be quite fruitful both as vehicles for research and as means of providing significant and practical natural language interfaces.

- **Command Interfaces to Operating Systems** -- Imperative command dialogs differ from data base queries in many important ways beyond the obvious differences in surface syntactic structure. But, much of the research on limited-domain semantics, ambiguity resolution, ellipsis and anaphora resolution can be exploited, extended and implemented in such domains. Moreover, there is no question as to the practical import and readily-available user community for such systems. What new linguistic phenomena do they highlight? More than one would expect. In our preliminary work leading up the the PLUME interface to the VMS operating system, we have found intersentential meta-language utterances, cross-party ellipsis and anaphora, and dynamic language redefinition, to name a few. An instance of intersentential meta-language typical to this domain would be:

```
USER: Copy foo.bar to my directory.  
SYST: File copied to [carbonell]foo.bar.  
USER: Oops, I meant to copy fum.bar.
```

There is no "oops command", nor any act for editing, re-executing, and undoing the effects of a prior utterance in the discourse. This is a phenomenon not heretofore analyzed, but one whose presence and significance was highlighted by the choice of application domain. See [2] for additional discussion of this topic.

- **Interfaces to expert systems** -- There is little question about the necessity, practicality and complexity of such a task. One can view expert systems as reactive, super data bases that require deduction in addition to simple retrieval. As such, the task of interpreting commands and providing answers is merely an extension of the familiar data-base retrieval scenario. However, much of the interesting human computer interaction with expert systems, as we discovered in our XCALIBUR interface [1], goes beyond this simple interaction. To wit, expert system interfaces require:

- Mixed-initiative communication, where the system must take the initiative in order to gather needed information from the user in a focused manner.
- Explanation generation, where the system must justify its conclusion in human-comprehensible terms, requiring user modelling and comparative analysis of multiple viable deduction paths.

- Knowledge acquisition, where information supplied in natural language must be translated and integrated into the internal workings of the system

- **Unified multi-function interfaces** -- Ideally one would desire communication with multiple "back ends" (expert systems, data bases, operating systems, utility packages, electronic mail systems, etc.) through a single uniform natural language interface. The integration of multiple discourse goals and need to transfer information across contexts and subtasks present an additional layer of problems -- mostly at the dialog structure level -- that are absent from interfaces to single-task, single-function backends.

The possible applications meeting the criteria have not by any means been enumerated exhaustively above. However, these reflect an initial set, most of which have received some attention of late from the computational linguistics community, and all appear to define theoretically and practically fruitful areas of research.

5. References

1. Carbonell, J. G., Boggs, W. M., Mauldin, M. L. and Anick, P. G., "The XCALIBUR Project, A Natural Language Interface to Expert Systems," *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, 1983.
2. Carbonell, J. G., "Meta-Language Utterances in Purposive Discourse," Tech. report, Carnegie-Mellon University, Computer Science Department, 1982.
3. Kaplan, S. J. and Davidson, J., "Interpreting Natural Language Data Base Updates," *Proceedings of the 19th Meeting of the Association for Computational Linguistics*, 1981.
4. Salvater, S., "Natural Language Data Base Update," Tech. report 84/001, Boston University, 1984.