An analysis of scripts generated in writing between users and computer consultants

by DAVID CHIN

University of California at Berkeley
Berkeley, California

ABSTRACT

The scripts generated in written interactive communications between users and a computer consultant program were investigated in a controlled experiment. The program was a simulation of UC, the UNIX Consultant, which users believed to be the actual program. An analysis of the scripts generated while solving a predefined set of problems showed the heavy use of context in forms such as ellipsis, anaphora, indirect speech acts, and grammatically incomplete sentences in over one-quarter of input clauses. Also present were grammatically ill-formed constructions and spelling errors. A comparison with a control group of users solving the same problem set with human consultants showed that the control group relied on context about twice as much as the simulation group. This suggests that people naturally use context in language and that the simulation group tried to rely less on context because they believed that they were speaking to a computer. Even so, contextual information is essential to understanding a large part of the simulation group's input.
INTRODUCTION

UC, the UNIX Consultant, is a large natural language interface under development at the University of California, Berkeley. The main goal of UC is to provide a natural help facility for naive users of the UNIX operating system. The user can converse with UC in English in the domain of UNIX and obtain advice on problems much as one would with a human consultant. The UC system is described at length in Wilensky, Arens, and Chin, and a brief overview can be found in Wilensky. Other aspects of UC are described in Arens, Chin, Jacobs, and Faletti.

During the development of UC, it was decided that it was necessary to test UC in a real setting to obtain actual performance requirements data for a system like UC. Although transcripts of user interactions with human consultants had been collected and used as models for UC, there was still a question of whether or not users would behave differently with a computer consultant. As with many large software projects, UC was not yet at the stage where field testing was possible, mostly because UC did not have sufficient knowledge to ensure a high enough hit ratio when users queried UC. Therefore, the usual solution of running a simulation of UC was tried, in this case, with actual human consultants who simulated UC in a controlled experiment.

In order to keep the scope of the experiment manageable, it was decided to focus on a single topic of interest, although the general procedure described in this paper is applicable to other areas. This experiment was designed mainly to evaluate how much users relied on contextual information in interactive written communications with a computer consultant. Currently UC is capable of handling a simple question/answer dialogue in the domain of the UNIX operating system. UC has some capabilities for handling contextual references, including anaphora, some elliptical constructs, and simple speech act analysis. There is a large effort underway to expand the capabilities of UC in these areas, with the ultimate aim of creating a version that will be able to carry on a coherent conversation with the user. Before embarking on such a large project, it was deemed advisable to determine if such additional capabilities would be useful.

THE EXPERIMENT

This experiment was designed to test for differences in language usage when users communicate with a computer consultant and when users communicate with human consultants. Because only the actual communications were of interest, only transcripts of experiment sessions were collected. However, similar procedures can be used to collect mental protocols, as defined by Lindsay and Norman, perhaps by having the subjects think aloud and using videotaping equipment as Lewis and Mack did. Also neglected in this experiment were timing information, task level analyses, and other human factors issues. A general introduction to such issues can be found in Card, Moran, and Newell.

Volunteer students from an introductory data structures course were enlisted to participate in an evaluation of UC. Six students were put through simulated UC sessions following a predefined problem set. Six additional students provided a control group and were told that they were writing to actual people. The instructions were in a written format to ensure uniformity and to avoid unplanned verbal biases. A sample of the instructions and problem set can be found in the Appendix.

The Simulation Group

In order to distract attention from the true aims of the experiment, participants were told when they were enlisted that they were there to test and evaluate the performance of the actual UC program. This perception was further corroborated by the instructions and the problem set, which asks for evaluations of ease of use for each problem and for an overall evaluation at the end of the session.

The Problem Set

The participants were expected to be at about an intermediate experience level in UNIX; therefore, the problems selected were for an intermediate to expert level. The aim was to design problems that most of the students had never encountered or that were obscure enough that the participants would be unsure of the solutions. Because the influence of the problem types and degree of difficulty might have on the scripts was unknown, the problems were designed to be new to the students so that they would not tailor their questions to what they considered the proper answers. In addition, the problems were worded in a format using the least possible information. This was done in order to approximate actual problems that users might encounter and to avoid biasing the scripts that the participants might use to communicate with UC. Finally, the problem set was designed to be a series of interrelated problems; it was felt that a cohesive set would be more typical of an actual session than a set of unrelated problems (there is no evidence for this "neglected" conjecture) and more important that such a cohesive set would provide opportunities for participants to use conversational context in their dialogue with UC.
Running the Simulation

The simulators were the expert implementors of UC. In order to better simulate UC, which sends the entire response at one time and which provides a prompt, a small program was written to simulate the interface. This program utilizes an emacs editor buffer to allow the simulators to edit the entire response before transmission. The actual transmission was done by the UNIX write utility. Simulators were also provided with a small number of frequent UC responses, including UC's query about misspelled or unknown words, UC's response to undecipherable input, and UC's typical response to questions about what UC knows. These responses were automated and bound to function keys for convenience and speed and to ensure uniformity of response within a session. The send key also automatically added UC's "#" prompt to the end of the transmission.

The Control Group

The second half of the experiment was a control. Six additional students were told that they were writing to actual people and that they were a part of the control for the previous experiment. These students were also given the same problem set as the simulation group and proceeded in much the same fashion as the simulation group. The only difference was that the control group were told that they were communicating with people. In several cases, the consultants to whom they were writing were in the same room.

Running the Control

The control group was run using the UNIX write command, which allows line-at-a-time communications between two terminals. Using write, a line is not sent to the receiving terminal until the return is hit. This allows users to correct mistakes on the same line before transmission. The sessions were recorded using the UNIX script command, which keeps a file copy of input/output for a terminal.

RESULTS

Of the six simulation participants, four considered themselves intermediates in UNIX experience, one was a beginner, and another was an advanced intermediate. The control group included four intermediates, one beginner, and two experts; however, the beginner and one of the experts combined to run one control session due to a shortage of terminals and time.

In the evaluations, the users consistently rated UC as a program they would use in learning UNIX and would recommend to friends who were starting out on UNIX. Although the users may have tried to avoid dependence on contextual information in their queries, the data shows that the attempts were certainly not very successful. More than one-fourth of all clauses used by the simulation group still required some knowledge of the context for a program like UC to be able to understand the clause. This means that any natural language program like UC would need to have such capabilities in order to be acceptable to the general public. Moreover, as the control group shows, "natural" conversation would require contextual understanding in over half the clauses.

slow. This means that UC will have to be much faster than its human equivalent to be acceptable.

Perusal of the transcripts from the simulation group shows about 7 cases of elliptical constructs used by the students, 12 cases of anaphora, 4 cases of words or sentences used only to maintain conversational coherence, 3 cases of grammatically ill-formed input, and 6 misspellings.

One of the control students did not believe that he was writing to a human, so that session was dropped from the statistical analysis. The other 5 sessions showed 18 cases of ellipsis, 19 cases of anaphoric references, 8 cases of conversational coherence constructs, 6 cases of grammatically ill-formed input, and 3 misspellings.

Normalizing the statistics from the simulation and the control groups to counts per 100 clauses (or counts per 1000 words) shows that the control group used context about twice as often as the simulation group. A summary of the results is presented in [Table I.]

<table>
<thead>
<tr>
<th>Construction</th>
<th>Simulation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>ellipsis</td>
<td>8</td>
<td>21</td>
</tr>
<tr>
<td>anaphora</td>
<td>13</td>
<td>22</td>
</tr>
<tr>
<td>conversational</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>ill-formed</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>total clauses</td>
<td>91</td>
<td>85</td>
</tr>
<tr>
<td>total words</td>
<td>668</td>
<td>615</td>
</tr>
</tbody>
</table>

CONCLUSIONS

The doubled frequency of usage of contextual information when participants believed themselves to be talking to actual human beings rather than a computer program seems to indicate that the simulation group was consciously or unconsciously trying to rely less on context than the control group did. This was most likely because of preconceived notions about what computers can and cannot understand. One student remarked with surprise in his evaluation that "UC" was able to remember the previous query and use that to understand the next question.

Although the students may have tried to avoid dependence on contextual information in their queries, the data shows that the attempts were certainly not very successful. More than one-fourth of all clauses used by the simulation group still required some knowledge of the context for a program like UC to be able to understand the clause. This means that any natural language program like UC would need to have such capabilities in order to be acceptable to the general public.
APPENDIX

Evaluation Form and Problem Set

1. Introduction

UC is an experimental AI project that is supposed to behave like an expert UNIX Consultant. The idea is that since UNIX (and other operating systems) are very cryptic to the beginning user, it would be useful to have a computer utility that could take the place of a human consultant. This program like its human counterpart should be able to answer questions and provide advice about UNIX in English. The UC system is at a point in development where we (the implementors—about 8 grad students in the Berkeley Artificial Intelligence Research group) desperately need experience with real use of the system and evaluations of its utility. This is where you come in.

2. Your Background

Just a few questions to help us establish your background.

How many years/months have you used Unix? ___ years and ____ months.

How would you describe yourself as a Unix user? (circle one) beginner intermediate expert

Please describe in words how often you have used Unix: For example: Only for courses cs153 and cs3. Worked for one summer as a C programmer. Have written various game programs and done extensive hacking.

3. Instructions

This is your session with UC. Do not look at what your neighbor is doing. If you have used/played with UC before, please let us know so that we can take that into account in analyzing your session. Pretend that you are a beginning Unix user. You have encountered the following problems and would like to get the answers from UC:

(a) You have an account on ucbory and you have just gotten a new account on another machine (ucbkim) on the ethernet (a high speed interconnection among different machines much like the old berknet). You would like to move some of your files from ucbory to your new account on uckkim. A friend has told you that there is a very easy way to do this, but you can't remember what the command(s) were. Since it is late at night and no one else is around, you decide to ask UC.

ease of use (circle one): 1 2 3 4 5
(1 is very hard, 5 is very easy)

(b) Now that you have the solution for the above problem, you realize that what you really want to do is copy whole directories over to your new account and maybe there is an easy way to do this.

ease of use (circle one): 1 2 3 4 5
(1 is very hard, 5 is very easy)

(c) One of the things you tried to do was to make a link to one of your friend's files. However, new machine, uckkim gave you the error message "/na/friend/foo: Cross-device link." Pretend that you have never seen that error before and ask UC about this problem.

ease of use (circle one): 1 2 3 4 5
(1 is very hard, 5 is very easy)

(d) You now go off to another terminal and try the command suggested by UC for copying files from one machine to another via the ethernet, but find out that it doesn't work. You get the error message "Login incorrect." So now you come back to this terminal and want to know why it didn't work.

ease of use (circle one): 1 2 3 4 5
(1 is very hard, 5 is very easy)

(e) This space is left for you to be creative and ask your own questions.

ease of use (circle one): 1 2 3 4 5
(1 is very hard, 5 is very easy)

4. Evaluation

If you were a new Unix user, would you use UC? yes no

If UC were available, would you recommend it to your friends who are getting started on Unix? yes no

If you had a home computer and a similar system were available for the operating system of your micro, would you buy it, and what price (in % of the computer system price) would you be willing to pay? (Note that this does not mean that UC-like systems are even close to commercial availability, for one thing, UC is much too big to fit on almost any home computer—this question is just to estimate the value of such a facility).

yes no if yes, then price __ %.

This space is left for you to make general comments/suggestions/criticisms. All your comments will be seriously considered and are very much appreciated.

REFERENCES


