From:MAICS-99 Proceedings. Copyright © 1999, AAAI (www.aaai.org). All rights reserved.

Speed and Accuracy using Four Boolean Query Systems

Michael Chui
Computer Science Department and Cognitive Science Program
Indiana University
Lindley Hall 215
Bloomington, Indiana 47405
<mchui@cs.indiana.edu>

Andrew Dillon
School of Library and Information Science
Indiana University
Bloomington, Indiana 47405
<adillon@indiana.edu>

Abstract

Existing research has shown that naive users find it difficult to correctly express queries using the standard Boolean connectives AND, OR, and NOT. We conducted an empirical study examining the query production performance of naive users on four different systems: One of these systems was non-procedural textual (Boolean connectives); another was procedural textual (If-Then-Else), another non-procedural graphical (based on Venn diagrams), and the last was procedural graphical (Filter-Flow). We found significant differences in both the accuracy and speed of users' query production based on the underlying Boolean forms of the queries they were required to generate. User performance in accuracy and speed was highest for the system using Boolean connectives, followed by Venn, Filter-Flow, and then If-Then-Else. No overall speed/accuracy tradeoff was observed. However, interactions were discovered amongst the factors of Boolean form, procedurality, and text vs. graphics. These results suggest lines of further work that can be used to improve Boolean interfaces to Web search engines.

Introduction

One important area of application for cognitive science research is in the design of representation systems, in particular, those that form the foundation for various types of human-computer interactions. A category of such interactions that has been empirically studied is the production of Boolean queries for information retrieval (IR) systems.

IR systems are systems which store documents that can be retrieved by matching them to user-defined queries (Korfage, 1997). Traditionally, such IR system have included databases of text documents, electronic card catalog systems in libraries, etc. While a number of other query systems, such as ranked retrieval, have been advocated as being easier to use than queries based on Boolean logic, IR systems based on Boolean queries are still one of the most prevalent class of IR systems (Frants et al, 1999).

Hence, over the years, there have been a number of experiments that have studied different systems of expressing queries, in order to address the difficulties that users have in producing queries based on Boolean logic. For example, Greenblatt and Waxman (1978) studied users' query production in three database query languages that varied along the dimension of procedurality, i.e., the degree of "step-by-step-ness" (Welty & Stemple, 1981). Two of the database query languages were less procedural than the third algebra-based language. They found that subjects using the algebra-based language formulated the most incorrect queries in the longest time. These results indicate that the non-procedural query languages allowed users to best formulate correct queries.

Welty and Stemple (1981) also tested users' ability to produce queries in two relational database query languages that differed primarily with respect to their procedurality. They found that subjects (with no prior programming experience) that used the more procedural language formulated significantly more accurate queries on difficult problems than subjects that used the non-procedural language.

Michard (1982) also compared two query systems based on Boolean logic that differed with respect to their procedurality. However, one was textual (TEST) and the other graphical (GQL). The less procedural system, GQL, was based on Venn diagrams. Michard found that users of the Venn-based system made one fourth the errors

that users of the more procedural, textual system made. In analyzing errors, Michard also found:

1. The AND operator seemed to be the easiest to use.

2. Differences in the number of errors between TEST and GQL could be explained by the misuse of parentheses.

3. Almost all subjects correctly formulated exclusive OR (XOR) queries using GQL, but few correctly formulated XOR queries using TEST.

Hence, the underlying Boolean form of a query also significantly affects the ability of a user to formulate a particular query. Greene et al (1990) studied the accuracy of Boolean query production in two database query languages. They found that subjects produced the most accurate queries when they only involved the operator "AND." They also report that "OR alone" queries were harder than "AND alone" queries. "NEGATION" queries were harder than "AND alone" or "OR alone" queries. "AND+OR" queries were harder than all other query types.

Young and Shneiderman studied query production in two systems that differed in terms of their procedurality. The more procedural system was graphical ("filter-flow") and the other was textual (SQL). Subjects formulated significantly more correct answers in the filter/flow condition than in the SQL condition. They reported that the simplest queries to form in both interfaces was the combination of three intersection operators and the negation of one of the terms. They also reported that a common error was for subjects to use "AND" instead of "OR."

These findings suggest that several task and interface variables are important determinants of query production performance: procedurality, form, and graphics vs. text. In addition, all of the cited studies tested users' abilities to formulate queries on relational databases. Despite the explosion in use of the World Wide Web, there is little published work on different query systems for searching the Web. Hence, we attempt to extend the work comparing various types of Boolean query systems into forms that could be implemented on the Web.

Method

Subjects

24 paid undergraduate subjects were recruited from telecommunications classes at Indiana University - Bloomington (17 female and 7 male). Subjects were required not to have studied Boolean logic, set theory, or Venn diagrams. They were also required not have had any computer programming experience. 3 male and 9 female subjects were assigned to the text group. 4 male and 8 female subjects were assigned to the graphical group.

Design

A three-factor (procedurality, form, text/graphics) experimental design with partial repeated measures was employed. Subjects were assigned randomly to either a text group or a graphical group. Each subject was then tested in both a procedural condition and a nonprocedural condition, formulating 16 queries in each condition. Each of the queries has one of eight underlying Boolean forms, e.g. A AND B AND C. The following matrix illustrates the four conditions under which users were tested, separated by the dimensions of procedurality and text/graphics.

	Procedural	**Non-procedural**
Text	If-Then-Else	Boolean
Graphics	Filter-Flow	Venn

Conditions

If-Then-Else If-Then-Else is a procedural, textual system for expressing Boolean queries. It is similar to a part of many programming languages. In this system, queries take the following form:

```
IF condition
  THEN result1
  ELSE result2
```

In an actual query, `condition` is replaced with a word that might be found on a Web page. `result1` and `result2` are either replaced with "ACCEPT," "REJECT," or another IF-THEN-ELSE statement. A query basically takes the form of a program written to examine each Web page in turn, and respond with an "ACCEPT" if the query should return the page, or a "REJECT" if the query should not return the page. For

example, the following query would return all of the Web pages that mentioned either "dog" or "cat" but not both (i.e., exclusive OR):

```
IF dog
  THEN IF cat
         THEN REJECT
         ELSE ACCEPT
  ELSE IF cat
         THEN ACCEPT
         ELSE REJECT
```

Boolean The system using Boolean connectives is non-procedural and textual. Queries in this system are formed from the connectives AND, OR, and NOT, as well as parentheses (Without parentheses, AND was considered to have higher precedence than OR.). For example, the following query is one that returns pages with either "dog" or "cat" but not both:

```
(dog OR cat) AND NOT (dog AND
cat)
```

Filter-Flow The Filter-Flow system used in this experiment is a procedural, graphical system modeled on the system described in Young & Shneiderman. A filter in this system is represented by a box with a word in it. Such a filter permits only those pages to pass through it that contain the word in the box. These filters are connected by lines with directional arrows. A filter may also be modified by the addition of a circle with an "X" through it, creating a "negative filter." Such a filter will only allow pages to pass through it that do not contain the word in the box. For example, the following query returns pages containing either "dog" or "cat" but not both:

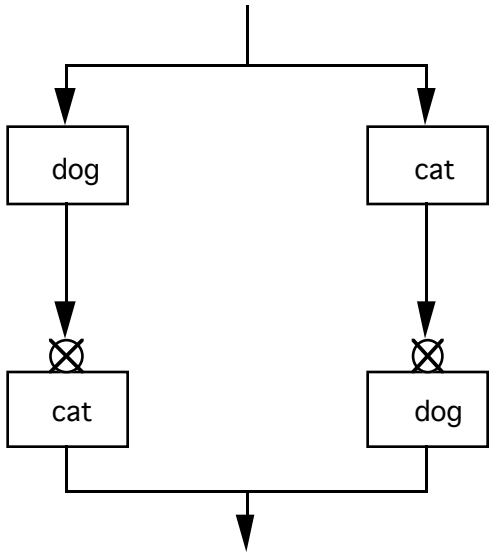


Figure 1: Filter-Flow XOR Example

Venn The Venn query system used in this experiment is a non-procedural, graphical system based on Venn diagrams. In this system, the set of all Web pages that contain a particular word is represented by a closed figure labelled by that word. Each such closed figure included in a query must overlap all of the other subregions created by other such closed figures. Regions in which two closed figures overlap represent Web pages that contain both of the words associated with the overlapping closed figures. Regions that are shaded by the subject represent those Web pages that will be returned by the query. For example, the following query will return the exclusive OR of pages containing "dog" and pages containing "cat":

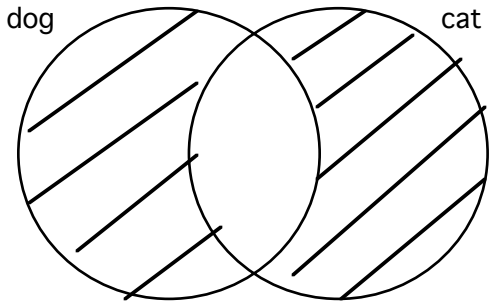


Figure 2: Venn XOR Example

While it is quite simple to draw a well-formed Venn diagram for one, two, or three terms, it is quite difficult to draw a Venn diagram for four terms. The number of subregions in a Venn diagram increases exponentially as a function of the number of terms (2^n-1 subregions for n

terms). Hence, a Venn diagram with four terms has 15 subregions:

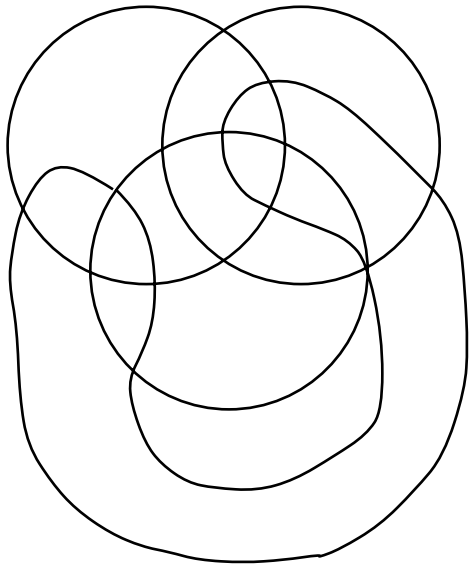


Figure 3: Venn 4-term Example

As a result of this difficulty, in this condition, subjects were provided with unlabelled diagrams containing the appropriate number of closed figures for the given queries. In a working on-line system, the computer would generate such diagrams. Subjects were required to label the closed figures, and shade in subregions.

Task

Each subject was tested in both procedural condition and a nonprocedural condition. These conditions were counterbalanced across subjects by order. For each condition, the subject was given a training session in which the query system was introduced. Subjects were shown how to express conjunction (AND), disjunction (OR), and negation (NOT). In the case of Boolean operators, precedence and parentheses were also introduced. Finally, a pre-test that required the subject to express an exclusive OR query was given. XOR was chosen because it requires the use of conjunction, disjunction, and negation. The subject was permitted, during training, to ask the experimenter for aid in formulating the query. After correctly completing the pre-test, and positively expressing an understanding of the query system, the actual trials began.

For each condition, subjects were required to formulate 16 queries. Each query was described in English at the top of a piece of paper, and space was provided underneath for the subject to write/draw the query using the system in which they had just been trained. The English query descriptions were formulated not to use either of the words "and" nor "or." Each of the 16 queries took one of eight Boolean forms (each Boolean form occurred twice). The order of these forms was randomized for each subject. The forms used are listed below:

```
1.A AND B AND C
2.A OR B OR C
3.A AND B AND NOT C
4.A AND (B OR C)
5.A OR (B AND C)
6.(A OR B) AND NOT C
7.(A AND B) OR (C AND D)
8.(A AND (B OR C)) OR D
```

Each trial was videotaped so that the time to complete each query could be determined.

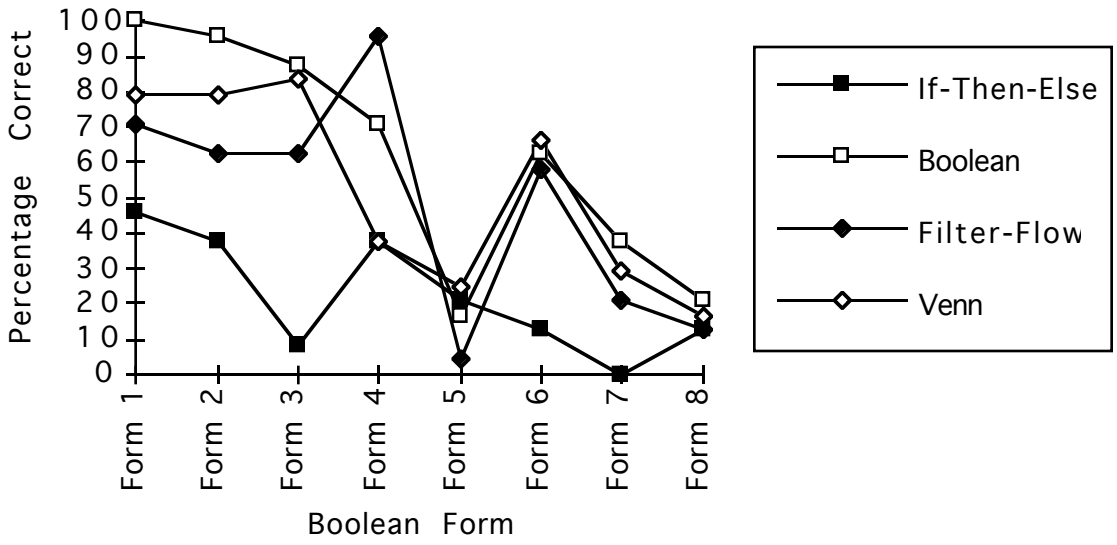
Results

We analyzed the accuracy of users' query production, i.e., the percentage of queries correctly formulated by subjects. Using a multivariate analysis of variance (adjusted for non-sphericity) on accuracy only, significant main effects were found for procedurality ($p=.001$) and form ($p<.001$), both within-subjects factors. Significant interactions were observed for procedurality * text/graphics ($p=.003$), procedurality * form ($p<.001$), and the three-way interaction procedurality * form * text/graphics ($p<.001$).

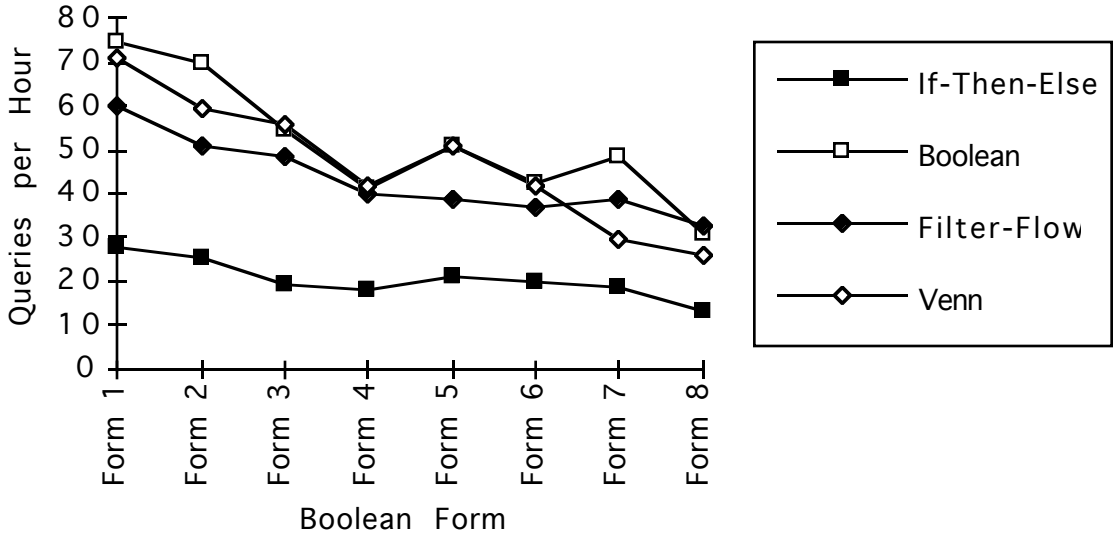
We also analyzed the speed of query production, i.e., the inverse of the amount of time that a user spent on a particular query. For speed only, significant main effects were found for procedurality ($p=.001$), form ($p<.001$), and text/graphics ($p=.030$). Significant interactions were observed for procedurality * text/graphics ($p<.001$), procedurality * form ($p<.001$), and the three-way interaction procedurality * form * text/graphics ($p=.013$).

Finally, we performed a doubly multivariate analysis of variance on speed and accuracy together. Significant main effects were found for procedurality ($p<.001$) and Boolean form ($p<.001$), both within-subjects factors. The between-subjects factor of text vs. graphics also was significant ($p=.036$). Significant two-way interactions were found for procedurality * text/graphics ($p<.001$) and procedurality * form ($p=.008$), but not for the three-way interaction.

Accuracy by Text/Graphics * Procedurality Form



Speed by Text/Graphics * Procedurality Form



Discussion

These results confirm that Boolean form is a very important factor in determining user performance in query production. Queries with a greater number of terms in the underlying Boolean form were produced with lower accuracy and at slower speeds. Queries that only contained conjunction were produced more accurately and at higher speeds. Queries that included negation were produced less accurately and at lower speeds than queries that did not contain negation.

We examined the possibility of a speed/accuracy tradeoff, i.e., subjects who spent more time on each task might have produced more accurate queries than subjects who spent less time on a task. Pearson correlations were run comparing the accuracy scores of individual subjects on each Boolean form and condition with the speed statistics on the same trials. No consistent significant negative (or positive) correlation was found. We conclude that our data does not show evidence of a speed/accuracy tradeoff on these tasks.

Accuracy scores on Boolean forms with a "global OR" structure (Form 5: A OR (B AND C), Form 7: (A AND B) OR (C AND D), Form 8: (A AND (B OR C)) OR D) were unexpectedly low. It is possible that this might simply be the result of a difficult wording of the English queries or a difficulty in reading comprehension. Alternatively, they might reflect a greater level of difficulty in the processing of disjunctive information. Johnson-Laird (1983) postulates that human processing of logical syllogisms is limited by the number of alternate models that can be simultaneously maintained in working memory. Disjunctive queries might require subjects to entertain several "models" of the required results, thus taxing the working memory capacity of searchers. Further elucidating the reasons for these results is an area for further study.

Interestingly, the speed with which subjects formulated queries on Form 5 in the Boolean (non-procedural textual) and Venn (non-procedural graphical) cases seems to be elevated while the accuracy with which subjects formulated those queries was much poorer compared to other queries with the same number of terms. Once again, attempting to discover if this is the result some kind of localized speed/accuracy tradeoff is an area for further study.

Accuracy was generally highest for the Boolean condition, followed by Venn, Filter-Flow, and then, with much lower scores, If-Then-Else. The same general pattern was found in the speed data. The relatively slow speed with which subjects formulated queries in the If-Then-Else (procedural textual) case can be explained in terms of the greater amount of writing required to record these queries. Similarly, the relatively faster speed with which subjects formulated queries in Venn can be partly explained by the fact that closed curves were already drawn for them.

While we attempted to control for prior experience by requiring subjects to state that they had not studied Boolean logic, set theory, or Venn diagrams, all of our subjects were American undergraduate university students. Because Boolean logic and Venn diagrams are widely taught in American schools, it is quite possible that some subjects were exposed to these systems previously. These variables should be better controlled in future studies.

From a cognitive perspective, it would be worthwhile to further examine the causes for both the decreased accuracy on "global OR" Boolean forms and the increased speed on Form 5 (A OR (B AND C)) in certain conditions. Johnson-Laird's mental models theory might explain decreased accuracy on disjunctive queries and with greater numbers of terms, but a theory explaining the differences in performance amongst different representational query systems has not yet been advanced.

From a design perspective, these results seem to show that our traditional Boolean query systems provide the most accurate and efficient method of formulating queries. However, even using this system, subjects performed very poorly on several of the Boolean forms, and in some cases, other representational systems allowed more accurate performances. Perhaps a further examination of these cases can provide insight into how to design systems that allow users to generate queries more accurately and efficiently.

A cognitive theory that can account for the various results from this study has not yet been developed. However, it is clear that such a theory would have to explain how the factors of procedurality, Boolean form, and text vs. graphics can affect both the accuracy and speed of query production. Such a theory would likely provide some guidance in how to design a query system that would result in more effective and efficient query production by end users.

References

1. Frants, V.I., Shapiro, J., Taksa, I., and Voiskunskii, V. (1999). Boolean Search: Current State and Perspectives. *Journal of the American Society for Information Science*, 50(1):86-95.

2. Greenblatt, D. and Waxman, J. (1978). A study of three database query languages. In Sheiderman, R., Ed. *Databases: Improving Usability and Responsiveness*. pp. 77-79. New York: Academic Press.
3. Greene, S., Devlin, S., Cannata, P. and Gomez, L. (1990). No IFs, ANDs, or ORs: A study of database querying. *International Journal of Man-Machine Studies*, 32, 303-326.
4. Johnson-Laird, P. (1983). *Mental Models*. Cambridge, MA: Harvard University Press.
5. Korfage, R. (1997). *Information Storage and Retrieval*. pp. 63-65. New York: John Wiley & Sons.
6. Michard, A. (1982). Graphical presentation of boolean expressions in a database query language: design notes and an ergonomic evaluation. *Behaviour and Information Technology*, 1(3):279-288.
7. Welty, C. and Stemple, D. (1981). Human Factors Comparison of a Procedural and a Nonprocedural Query Language. *A C M Transactions on Database Systems*, 6, 626-649.
8. Young, D. and Shneiderman, B. (1993). A Graphical Filter/Flow Representation of Boolean Queries: A Prototype Implementation and Evaluation. *Journal of the American Society for Information Science*, 44(6):327-339.