A COMPARISON OF UNCERTAINTY CALCULI IN AN EXPERT SYSTEM FOR INFORMATION RETRIEVAL

Richard M. Tong, Daniel G. Shapiro,
Jeffrey S. Dean and Brian P. McCune

Advanced Information & Decision Systems
201 San Antonio Circle, Suite 286
Mountain View, CA 94040, USA.

ABSTRACT

This paper describes the first stages of an experimental investigation into the effects of using various calculi to propagate uncertainty in an interactive expert system for information retrieval. We interpret uncertainty values as partial truths rather than probabilities, and draw upon the mathematics of multi-valued logic in developing our analysis. We conclude that specification of an uncertainty calculus is a subtle problem that interacts in several ways with the scheme used to represent the expert knowledge.

I INTRODUCTION

In this paper we report on an experiment intended as the first in a series designed to explore the effects of using different representations of uncertainty within a rule-based expert system. Existing systems employ a variety of uncertainty calculi which although based on formal theories, are usually implemented in an ad hoc way with little or no effort expended on experimental tests of their validity.

In an attempt to clarify some of these issues, we have used RUBRIC (RUle Based Retrieval of Information by Computer), a research prototype system for rule-based information retrieval (see McCune et al [2] for full system details), as the vehicle for our investigation. Information Retrieval is a good domain for such experimentation since the user is responsible for both the knowledge base and the ground truth against which performance is measured. The RUBRIC system is designed to help users by providing automated and relevant access to unformatted textual databases. A specific retrieval request is carried out by a goal-oriented inference process, in which the root node of the search tree represents a semantic concept or topic that the user wants retrieved. Nodes further down the tree represent intermediate concepts with which the root is defined, and the nodes at the leaves of the tree represent patterns of words that are to be searched for in the database. Each arc in the tree may be given a weight, which ve can interpret as "truth" or "belief" or "confidence" as we wish. This allows the intermediate concepts and keyword expressions that are found to add differing amounts to our overall confidence that the root concept has indeed been retrieved. It is with the calculus by which these uncertainty values are propagated that this paper is concerned.

II EXPERIMENTAL METHOD

Within the literature of expert systems, there have been several attempts to construct a "calculus of uncertainty", some based on the concepts of probability and others on the more general formalisms of mathematical logic (see Shafer[4] and Zadeh[6] for an introduction to some of these). In this experiment, we assume that uncertainty can be represented as a numerical value in the interval [0,1], but rather than view the uncertainty values as probabilities (as do systems such *as* PROSPECTOR), we view them as the truths of the associated propositions. That is, the uncertainty value attached to the proposition "x is A" is the truth of what it asserts rather than the probability that the event that it describes occurred. This being the case, we need to construct a calculus to handle these non-classical truth values. Fortunately, such calculi have been studied extensively [3] and go under the name of Multi-Valued Logics. We draw upon this work in what follows.

The first task is to define a set of operators for conjunction (the and connective), and disjunction (the or_ connective). There are many we could choose, but we shall consider three pairs as summarized in Table 1. Here v(A) and v(B) denote the truth values of the primary propositions, with v(A and B) and v(A or_ B) denoting the value of their conjunction and disjunction respectively. Negation (the unary operator not) is assumed always to be given by v(not A) ■ 1 - v(A).

The second task is to define a mechanism for performing rule-based inference. Recall that in two-valued logic the modus ponens rule allows B to be inferred from A and A->B. However, in a multi-valued logic, we need to extend this idea so that v(B) can be computed from any given v(A) and v(A->B), where -> is some multi-valued implication. Functions that allow us to compute v(B) are called detachment operators (and are denoted * in this paper). Again, there are many we could choose, but we have selected four which are shown in Table 2, together with the corresponding implications.

Let us denote a particular calculus by c(i,j) where 'i' is an index over the conjunct-disjunct

operators and 'j' is an index over the detachment operators. We see that some of the c(i,j) are well known; in particular, c(3,4) is Lukasiewicz's nondenumerably infinite system [l], and c(3,l) is a system proposed by Zadeh [5]. Another calculus of interest is c(2,3), which we can view as a '•pseudo-probability" logic in which A and B are independent events.

Table I. Conjunct-Disjunct Operators

	v(A and B)	v(A or B)
1	max[0,v(A)+v(B)-1]	min[1,v(A)+v(B)]
2	v(A).v(B)	v(A)+v(B)-v(A).v(B)
3	min[v(A),v(B)]	max[v(A),v(B)]

Table 2. Detachment Operators

	Detachment (*)	Implication (=>)
1	min[v(A),v(A=>B)]	min[v(A),v(B)]
2	min[v(A),v(A=>B)] if v(A)+v(A=>B) > 1 0 otherwise	max[1-v(A),v(B)]
3	v(A).v(A=>B)	min[1,v(B)/v(A)]
4	max[0,v(A)+v(A=>B)-1)]	min[1,1-v(A)+v(B)]

Having defined the operators, our basic experiment involves the definition of a query (i.e., a set of production rules), the selection of a representative story set and the repeated application of the query to the story set. Since we have defined twelve separate calculi for uncertainty propagation (three pairs of conjunct-disjunct operators and four detachment operators) the experiment will result in twelve potentially distinct orderings of the story set.

As a typical query we selected "Acts of Terrorism", and then developed a structure for this concept. This tree of successively more precise sub-concepts should be interpreted as a definition for a prototypical story, and is translated into a set of LISP production rules, some of which are shown in Figure 1. The first four rules in the query have consequents that correspond to the top node of the concept tree. When we compute the overall truth for a particular story, we combine the truths from these rules by using an or connective. At the lowest level in the tree antecedents to rules are simple keywords or keyword expressions.

```
(
 ( IMPLIES terrorism actor .2)
 ( IMPLIES terrorism event .7)
 ( IMPLIES terrorism effect .3)
 ( IMPLIES terrorism reason .1)

 ( IMPLIES reason "overthrow" .2)


 ( IMPLIES general-actor "terrorist" .6)
 ( IMPLIES general-actor "guerilla" .1)

 ( IMPLIES killing (*OR* shooting slaying) 1.0)

 ( IMPLIES bombing (*AND* device explosion) .8)
 ( IMPLIES bombing (*OR* device explosion) .6)
 ( IMPLIES device "bomb" .6)

 ( IMPLIES effect (*OR* "injure" "maim") .4)
 ( IMPLIES effect (*OR* "dead" "death") .5)

 ( IMPLIES effect "victim" .4)
)
```

Figure 1. LISP Implementation of Query (partial)

We then selected a set of thirty stories taken from the Reuters wire service as representative of the data that the RUBRIC system would encounter. A one-line summary of these is given in Figure 2. Notice that they all report some kind of violent activity but not all are relevant to our query. Those marked with a double asterisk were determined to be definitely relevant and those marked with a single asterisk were determined to be marginally relevant, thus defining our subjective ground truth.

III MEASURES OF PERFORMANCE

RUBRIC'S basic task is to assign a weight to each story in the data base. This weight is the truth of the statement "this story is relevant to the query", with its value being determined by propagating the uncertainty values through the structure defined by the query rule set. This makes the assessment of performance somewhat complicated, since we are interested in the properties of the ordering, both in absolute terms (i.e., the truth values returned) and with reference to the ordering that we determined beforehand.

```
     1  Overview story about the war in Chad and its effects.
     2  Overview story of the situation in Poland and Solidarity.
**   3  Short on car bomb in London.
*    4  UK deports Palestinian terrorist to Israel.
     5  FBI takes Reagan's secret code card after assassination attempt
     6  Political effects of attack on Angola's oil refinery.
**   7  Chilean secret service agent brought nerve gas into US.
**   8  Follow-on to London bombing story.
**   9  More on London bombing.
    10  Boxing match - WBC featherweight champion.
    11  Earthquake in Pakistan.
    12  Cyclone in India.
    13  Soviet reaction to Polish crisis.
    14  Reaction of Soviet bloc countries to Polish crisis.
    15  Spanish army officers placed under house arrest.
    16  Story on Iraq-Iran conflict.
*   17  Accidental chain of explosions at Army arms dump in Zimbabwe.
    18  General interest story about Napoleon and Waterloo.
**  19  Bomb explosions in two Yugoslav restaurants.
    20  Accidental explosion in apartment building in NE. Italy.
**  21  Italian couple freed by kidnappers after ransom paid.
**  22  Bomb explosion in central Tehran street.
**  23  Part story about murdered Italian industrialist.
    24  Iranian leftists executed by firing squad in Tehran.
    25  Shell exploded and killed bomb disposal experts in E. Beirut.
**  26  Mayor and seven others kidnapped and shot in Guatemala.
    27  Lawyers for Sadat's assassins argue against charges.
    28  Violence caused by Haitian refugees in Miami detention center.
**  29  Iranian Parliament member assassinated in Tehran.
    30  Brazilian athlete recovering from auto accident.
```

Figure 2. Summary of Reuters Stories

For the purposes of this paper we have adopted two basic measures. Both of these are based on the idea of using a selection threshold to partition the ordered stories so that those above it are "relevant" and those below it are "irrelevant". In the first we lower the threshold until we include all those deemed a. <u>priori</u> relevant, and then count the number of unwanted stories that are also selected (denoted N_F). In the second we raise the threshold until we exclude all irrelevant stories, and then count the number of relevant ones -that are not selected (denoted N_M). The first definition

$$n$$

therefore gives us an insight into the system's ability to reject unwanted stories (precision), whereas second gives us insight into the systems ability to select relevant stories (recall). There are other measures we could use to give a more complete picture of performance (see [2] for details), and we recognize that in practice the "goodness" of the system's performance will depend on a balance of these measures.

IV <u>ANALYSIS</u> Of. <u>RESULTS</u>

Using the two basic performance measures described in the previous section, RUBRIC'S performance is summarized in Table 3. For example, c(l,2) gave 6 false hits when we applied the precision measure and 9 missed stories when we applied the recall measure. Remembering that our story data-base has 30 entries of which 12 are marked as being relevant, we see that some calculi gave good performance whichever measure we used, while others performed veil on one but inadequately on the other. Interestingly, no one calculus seems to be significantly better than any of the others. Indeed, there seem to be four calculi, c(l,3), c(2,2), c(2,3) and c(3,3), whose performance is practically indistinguishable.

These results are most interesting. They show that a change in calculus can indeed have a marked effect on the interpretation of a query. Thus although some calculi seem totally inappropriate, there are others which apparently capture our notion of uncertainty. However, in our search for an understanding of why some calculi did better than others we became aware of the fact that there are some subtle issues that can affect the interpretation of our results. We discuss two of these in the next section.

Table 3. System Performance

	1	2	3	4
1	7	6	(4	6
2	7	(5	3	18
3	4	4	6)	18

Precision Scores
(N_F when N_M=0)

	1	2	3	4
1	12	9	(5	7
2	10	(4	4	5
3	11	11	2)	5

Recall Scores
(N_M when N_F=0)

Row indices denote disjunct-conjunct operator pairs. Column indices denote detachment operators. (See Tables 1 and *2.)* Scores for the better calculi are encircled.

V INCONSISTENCY AND DEPENDENCY

The first confounding effect we have called inconsistency, and it relates to the mismatch between two translations of the retrieval concept that we have used; we define both a prior ordering of the story set, which we can view as a declarative statement of the concept, and a rule-based query, which we can view as a procedural definition of the concept. Obviously, one or the other (or both) of these may not capture exactly the user's internal model of the concept being retrieved. If that is the case, then attempts to compare them will lead to errors of assessment. As examples of mis-translation of the first kind (i.e., incorrect labelling), consider stories <24> and <25>. Since these deal with terrorist-related acts in the Middle-East, they should probably be considered at least marginally relevant to the query, yet initially we considered them to be of no interest. As examples of the second kind (i.e., inadequate specification), consider stories <9> and <10>. While story <9> is definitely relevant (a story about a car bombing), it is often not selected when the query is applied to the story data base. On the other hand, story <10> is ju6t as clearly irrelevant (a story about a boxing match), and yet it often receives a high rating.

The second confounding effect we have called dependency. It is caused by interaction between the rule-based query and c(i,j). An implicit assumption *of* our experiment is that we can consider the effects of c(i,j) independently of the specification of the query. However, a particular query may rely on a particular c(i,j) for its effectiveness. Indeed, it could be that certain forms of query are basically incompatible with certain c(i,j). Thus, for example, if we wished to rely on the implicit disjunction between rules to produce a magnifying effect when there are several paths to a sub-concept, then a calculus which uses max-min as and-or connectives will never achieve the desired effect. So although calculus c(3,4) was not selected as one of the "best", this may be because the form of query we used mediated against it. Perhaps if we had fixed c(3,4) as our uncertainty calculus, then we could have constructed an effective query around it.

VI CONCLUSIONS

The experiment described above shows that changing the uncertainty calculus does change the performance of RUBRIC, with some of the calculi clearly failing to produce satisfactory results. However, the variations we observed had some unexpected characteristics which led us to a consideration of some deeper issues of consistency and dependency.

Our conclusion is that specification of an uncertainty representation is a problem of some complexity and subtlety. We believe that there are interactions between the form of the query and the admissible representations, and will continue to explore these, and other effects, in subsequent experiments.

REFERENCES

[1] Lukasiewicz, J. "Many-valued Systems of Propositional Logic" In McCall, S. Polish Logic. O.U.P., 1957.

[2] McCune, B.P., J.S. Dean, D.G. Shapiro, R.M. Tong "Rule-Based Retrieval of Information by Computer," Final Technical Report, TR-1018-1, Advanced Information & Decision Systems, Mtn. View, CA., January 1983.

[3] Rescher, N. Many Valued Logic. McGraw-Hill, New York, 1969.

[4] Shafer, G. A Mathematical Theory of Evidence. Princeton Univ. Press, 1976.

[5] Zadeh, L.A. "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes." IEEE Trans. Systems. Man, and Cybernetics. SMC-3:1 (1973) 28-44"

[6] Zadeh, L.A. "Approximate Reasoning Based on Fuzzy Logic" In Proc. IJCAI-79 Tokyo, Japan, August, 1979, pp. 1004-1010.