A New Frontier in Computation—Computation with Information Described in Natural Language

L. Zadeh

I. Extended Abstract

What is meant by Computation with Information Described in Natural Language, or NL-Computation, for short? Does NL-Computation constitute a new frontier in computation? Do existing bivalent-logic-based approaches to natural language processing provide a basis for NL-Computation? What are the basic concepts and ideas which underlie NL-Computation? These are some of the issues which are addressed in the following.

What is computation with information described in natural language? Here are simple examples. I am planning to drive from Berkeley to Santa Barbara, with stopover for lunch in Monterey. It is about 10 am. It will probably take me about two hours to get to Monterey and about an hour to have lunch. From Monterey, it will probably take me about five hours to get to Santa Barbara. What is the probability that I will arrive in Santa Barbara before about six pm? Another simple example: A box contains about twenty balls of various sizes. Most are large. What is the number of small balls? What is the probability that a ball drawn at random is neither small nor large? Another example: A function, f, from reals to reals is described as: If X is small then Y is small; if X is medium then Y is large; if X is large then Y is small. What is the maximum of f? Another example: Usually the temperature is not very low, and usually the temperature is not very high. What is the average temperature? Another example: Usually most United Airlines flights from San Francisco leave on time. What is the probability that my flight will be delayed?

Computation with information described in natural language is closely related to Computing with Words. NL-Computation is of intrinsic importance because much of human knowledge is described in natural language. This is particularly true in such fields as economics, data mining, systems engineering, risk assessment and emergency management. It is safe to predict that as we move further into the age of machine intelligence and mechanized decision-making, NL-Computation will grow in visibility and importance.

Computation with information described in natural language cannot be dealt with through the use of machinery of natural language processing. The problem is semantic imprecision of natural languages. More specifically, a natural language is basically a system for describing perceptions. Perceptions are intrinsically imprecise, reflecting the bounded ability of sensory organs, and ultimately the brain, to resolve detail and store, information. Semantic imprecision of natural languages is a concomitant of imprecision of perceptions.

Our approach to NL-Computation centers on what is referred to as generalized-constraint-based computation, or GC-Computation for short. A fundamental thesis, which underlies NL-Computation, is that information may be interpreted as a generalized constraint. A generalized constraint is expressed as X isr R, where X is the constrained variable, R is a constraining relation and r is an indexical variable which defines the way in which R constrains X. The principal constraints are possibilistic, veristic, probabilistic, usuality, random set, fuzzy graph and group. Generalized constraints may be combined, qualified, propagated, and counter propagated, generating what is called the Generalized Constraint Language, GCL. The key underlying idea is that information conveyed by a proposition may be represented as a generalized constraint, that is, as an element of GCL.

In our approach, NL-Computation involves three modules: (a) Precisiation module; (b) Protoform module; and (c) Computation module. The meaning of an element of a natural language, NL, is precisiated through translation into GCL and is expressed as a generalized constraint. An object of precisiation, p, is referred to as precisiend, and the result of precisiation, p*, is called a precisiand. Usually, a precisiend is a proposition, a system of propositions or a concept. A precisiend may have many precisiands. Definition is a form of precisiation. A precisiand may be viewed as a model of meaning. The degree to which the intension (attribute-based meaning) of p* approximates to that of p is referred to as cointension. A precisiand, p*, is cointensive if its cointension with p is high, that is, if p* is a good model of meaning of p.

The Protoform module serves as an interface between Precisiation and Computation modules. Basically, its function is that of abstraction and summarization.

The Computation module serves to deduce an answer to a query, q. The first step is precisiation of q, with precisiated query, q*, expressed as a function of n variables u1, …, un. The second step involves precisiation of query-

---

Department of EECS, University of California, Berkeley, CA 94720-1776; Telephone: 510-642-4959; Fax: 510-642-1712;

E-Mail: zadeh@eecs.berkeley.edu Research supported in part by ONR N00014-02-1-0294, BT Grant CT1080028046, Omron Grant, Tekes Grant, Chevron Texaco Grant and the BISC Program of UC Berkeley

relevant information, leading to a precisiand which is expressed as a generalized constraint on u1, …, un. The third step involves an application of the extension principle, which has the effect of propagating the generalized constraint on u1, …, un to a generalized constraint on the precisiated query, q*. Finally, the constrained q* is interpreted as the answer to the query and is retranslated into natural language.

The generalized-constraint-based computational approach to NL-Computation opens the door to a wide-ranging enlargement of the role of natural languages in scientific theories. Particularly important application areas are decision-making with information described in natural language, economics, systems engineering, risk assessment, qualitative systems analysis, search, question-answering and theories of evidence.