Semantic Acquisition In TELI: A Transportable, User-Customized Natural Language Processor

Bruce W. Ballard
Douglas E. Stumberger

AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974

Abstract

We discuss ways of allowing the users of a natural language processor to define, examine, and modify the definitions of any domain-specific words or phrases known to the system. An implementation of this work forms a critical portion of the knowledge acquisition component of our Transportable English-Language Interface (TELI), which answers English questions about tabular (first normal-form) data files and runs on a Symbolics Lisp Machine. However, our techniques enable the design of customization modules that are largely independent of the syntactic and retrieval components of the specific system they supply information to. In addition to its obvious practical value, this area of research is important because it requires careful attention to the *formalisms* used by a natural language system and to the *interactions* among the modules based on those formalisms.

1. Introduction

In constructing the Transportable English-Language Interface system (TELI), we have sought to respond to problems of both an applied and a scientific nature. Concerning the *applied* side of computational linguistics, we seek to redress the fact that many natural language prototypes, despite their sophistication and even their robustness, have fallen into disuse because of failures (1) to make known to users exactly *what inputs* are allowed (e.g. what words and phrases are defined) and (2) to provide capabilities that meet the *precise needs* of a given user or group of users (e.g. appropriate vocabulary, syntax, and semantics). Since experience has shown that neither users nor system designers can predict in advance all the words, phrases, and associated meanings that will arise in accessing a given database (cf. Tennant, 1979), we have sought to make TELI "transportable" in an extreme sense, where customizations may be performed (1) *by end users*, as opposed to the system designers, and (2) *at any time* during the processing of English sentences, rather than requiring a complete customization before English processing may occur.

In addition to the potential practical benefits of a user-customized interface, we feel that well-conceived transportability projects can make useful *scientific* contributions to computational linguistics since single-domain systems and, to a lesser extent, systems adapted over weeks or months by their designers, afford opportunities to circumvent, rather than squarely address, important issues concerning (a) the precise nature of the *formalisms* the system is designed around, and (b) the *interactions* among system modules. Although customization efforts offer no guarantee against ad-hoc design or sloppy implementation, problems of the type mentioned above are less likely to go unnoticed when dealing with a system whose domain-specific information is supplied at run-time, especially when that information is being provided by the actual users of the system.

By way of overview, we note that the TELI system derives from previous work on the LDC project, as documented in Ballard (1982), Ballard (1984), Ballard, Lusth and Tinkham (1984), and Ballard and Tinkham (1984). The initial prototype of TELI, which runs on a Symbolics Lisp Machine, is designed to answer English questions about information stored in one or more tables, (i.e. first-normal-form relational database). A sample view of the display screen during a session with TELI, which may give the flavor of how the system operates, is shown in Figure 1. Information on some aspects of knowledge acquisition not discussed in this paper, particularly with regard to syntactic case frames, can be found in Ballard (1986).

2. Types of Modifiers Available in TELI

The syntactic and semantic models adopted for TELI are intended to provide a unified treatment of a broad and extendible class of word and phrase types. By providing for an "extendible" class of constructs, we make the knowledge acquisition module of TELI independent of the natural language portion of the system, whose earlier version has been described in Ballard and Tinkham (1984) and Ballard, Lusth, and

Tinkham (1984). In the remainder of this paper, the reader should bear in mind that the acquisition modules of TELI, including the menus they generate, are driven by extensible data structures that convey the linguistic coverage of the underlying natural language processor (NLP) for which information is being acquired. For example, incorporating adjective phrases into the system involved adding 12 lines of Lisp-like data specifications. This brevity is largely due to the use of case frames that embody dynamically alterable selectional restrictions (Ballard, 1986).

As an initial feeling for the coverage of the NLP for which information is currently acquired, TELI provides semantics for the *word* categories

Adjective
e.g. an *expensive* restaurant
Noun Modifier
e.g. a *graduate* student
Noun
e.g. a *pub*

and the *phrase* types

Adjective Phrase
e.g. employees *responsible for* the planning projects
Noun-Modifier Phrase
e.g. the *speech* researchers
Prepositional Phrase
e.g. the trails *on* the Franconia-Region map
Verb Phrase
e.g. employees that *report to* Brachman
Functional Noun Phrase
e.g. the *size* of department 11387,
the *colleagues* of Litman

In addition to these user-defined modifier types, the system currently provides for negation, comparative and superlative forms of adjectives, possessives, and ordinals. Among the grammatical features supported are passives for verbs, reduced relatives for prepositional and adjective phrases, fronting of verb phrase complements, and other minor features. One important area for expansion involves quantifiers, both logical (e.g. "all") and numerical (e.g. "at least 3").

3. Principles Behind Semantic Acquisition

As noted above, our goal is to devise techniques that enable *end users* of a natural language processor to furnish all domain-specific information to by the system. This information includes (1) the *vocabulary* needed for the data at hand; (2) various types of *selectional restrictions* that define acceptable phrase attachments; and most critically (3) the *definitions* of words and phrases. With this in mind, the primary criteria which the semantic acquisition component of TELI has been designed around are as follows.

To allow users to define, examine or modify domain-specific information at any time. This derives from our beliefs that the needs of a user or group of users cannot all be predicted in advance, and will probably change once the system has begun operation.

To enable users to impart new concepts to the system. We provide more than just synonym and paraphrase capabilities and, in fact, definitions may be arbitrarily complex, by being defined either (a) in terms of other definitions, which may be defined upon other definitions, or (b) as the conjuction of an arbitrary number of constraints.

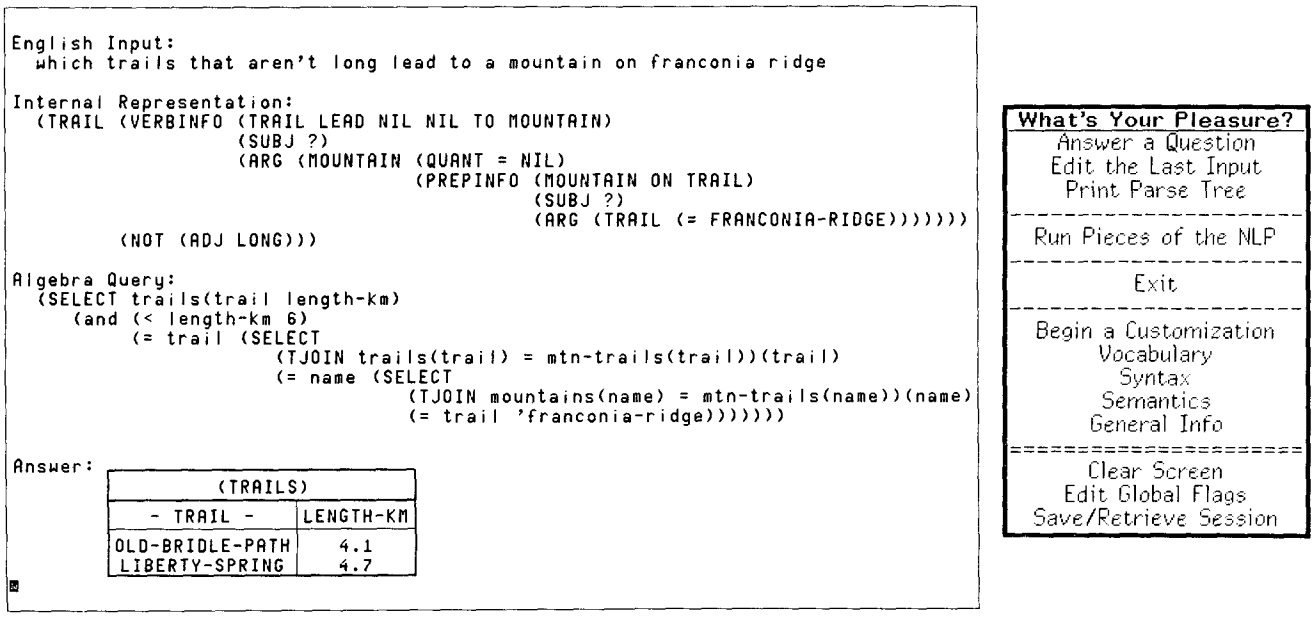

Figure 1: Sample Display Screen; Top-Level Menu of TELI

To provide definition capabilities independent of modifier type. In our system, adjectives, nouns, prepositional phrases, verb phrases, and so forth are all defined in precisely the same way. This is achieved in part by treating all modifiers as n-place predicates.

To allow definitions to be given at various conceptual levels. Users are able to specify meanings (a) in English; (b) in terms of the meanings of previously defined words or phrases; (c) by reference to "conceptual" relationships, which have been abstracted to a level above that of the physical data files; or (d) in terms of database columns. We strive to minimize the need for low-level database references, since this helps (1) to avoid tedious and redundant references, and (2) to assure that most of our techniques will be applicable beyond the current conventional database setting.

To provide alternate modalities of specification. For example, the menu scheme described in Section 7.2 offers the user more assistance in making definitions, but is less powerful, than the alternative English and English-like methods described in Section 7.3. We prefer to let users decide when each modality is appropriate, rather than force a compromise among simplicity, reliability, and power.

To enable the system to provide help or guidance to the customizer. When defining a modifier, users may view all current modifiers of, or functions associated with, the object type(s) in question. Many other opportunities exist for co-operation on the part of the system. To avoid unnecessary limitations, however, users are generally able to override any hints made by the system.

4. Semantic Processing in TELI

The semantic model developed for TELI, in which definitions are acquired from users, assumes that (1) modifier meanings will be purely extensional, and can thus be treated as n-place predicates, and (2) semantic analysis will be almost entirely compositional. Concerning the latter assumption, we note that (a) some important disambiguations, including problems of word sense, will have been made *during parsing* by reference to selectional restrictions (Ballard and Tinkham, 1984), and (b) minimal re-ordering does occur in converting parse trees into internal representations.

4.1 Types of Semantics

All user-defined semantics, however acquired, are stored in a global Lisp structure indexed by the word or phrase being defined. *Single-word modifiers* are indexed by the word being defined, its part of speech, and the entity it modifies; *phrasal modifiers* are indexed by the phrase type and the associated case frame. For example, the internal references

```
(new adj room)
(prep-ph (restaurant in county))
```

respectively index the definitions of "new", when used as an adjective modifier of rooms, and "in", as it relates restaurants to counties. As suggested by this indexing scheme, word meanings arise only in the context of their occurrence, never in isolation. Thus, "new room" and "restaurant in county" receive definitions, not "new" or "in". This decision lends generality to the definitional scheme, and any additional effort thereby needed to make multiple definitions is minimized by the provisions for borrowed meanings, as described in Section 7.4.

Although our representation strategies allow for definitions that involve relatively elaborate traversals of the physical data files, TELI does not presently provide for arithmetic computations. Thus, the input "Which restaurants are within 3 blocks of China Gardens?" requires a 2-place "distance" function and, unless the underlying data files provide distances between restaurants (there are N-squared such distances to account for), the necessary semantics cannot be supplied.

4.2 Internal Representations

As an example of the "internal representation" (IR) of an input, which results from a recursive traversal of a completed parse tree, and which illustrates preparations for compositional analysis, the (artificially complex) input

"Which Mexican restaurants in the largest city other than New Providence that are not expensive are open for lunch?"

will have [roughly] the internal representation

```
(restaurant (not (adj expensive))
        (nounmod-ph (food restaurant)
                    (nounmod (food (= Mexican)))
                    (head ?))
        (prep-ph (restaurant in city)
                 (subj ?)
                 (arg (city (super large)
                            (!= New-Providence))))
        (adj-ph (restaurant open for meal)
                (subj ?)
                (arg (meal (= lunch)))))
```

This top-level interpretation of the input instructs the system to find all restaurants that satisfy (a) the *negation* of the *1-place predicate* associated with "expensive", and (b) the three *2-place predicates* associated with the noun-noun, prepositional, and adjective phrases. Note that modifiers associated with

phrasal modifiers are referenced by their case frame, e.g. "restaurant in city". Within the scope of these references, case labels (e.g. "subj" and "arg") indicate which slots have been instantiated and which slot has been relativized, the latter denoted by "?". The list of slot names associated with each phrase type is stored globally. In most instances, the argument of a case slot can be an arbitrary IR structure, in keeping with the recursive nature of the English inputs being recognized.

Since IR structures are built around the word and phrase types of the English being dealt with, and since the meanings of words and phrases are stored globally, IR structures should not be regarded as a "knowledge representation" in the sense of KL-ONE, logical form, and so forth. Systems similar in goals to TELI but which revolve around logical form include TEAM (Grosz, 1983; Grosz, Appelt, Martin, and Pereira 1985), IRUS (Bates and Bobrow, 1983; Bates, Moser, and Stallard 1984), and TQA (Plath, 1976; Damerau, 1985). One system similar to TELI in building intermediate structures that contain references to language-specific concepts is DATALOG (Hafner and Godden, 1985).

5. The Initial Phase of Customization

When a user asks TELI to begin learning about a new domain, the system spends from five to thirty minutes, depending on the complexity of the application, obtaining basic information about each table in the the database (see Figure 2). Users are first asked to give the *key column* of the table. This information is used primarily to guide the system in inferring the semantics of certain noun-noun and "of"-based patterns. Next, users are asked which columns contain *entity* values as opposed to *property* values. Typical properties are "size", "color", and "length", which differ from entities in that (a) their values do not appear as an argument to arbitrary verbs and prepositions (e.g. other than "have", "with", etc.) and (b) they will not themselves have properties associated with them. Finally, users are asked to specify the *type* of value each column contains. This information allows subsequent references to concepts (e.g. "color") rather than physical column names. It also aids the system in forming subsequent suggestions to the user (e.g. defaults that can be overridden).

Having obtained the information indicated above, the system constructs definitions that allow simple questions to be answered, such as

"What is Sally's social security number?"
"What is the age of John"

Along with information freely volunteered by the user, these definitions can be subsequently examined or changed at the user's request.

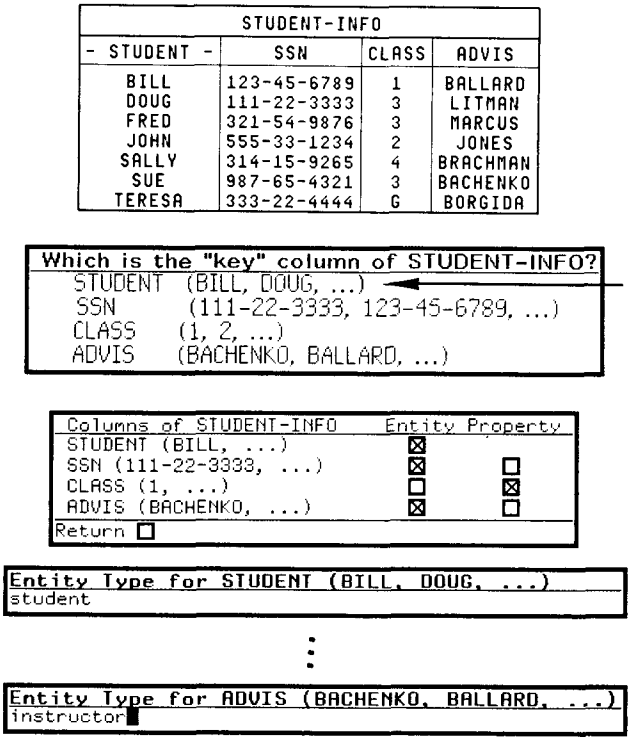

Figure 2: Initial Acquisitions

Based upon the answers to the questions described above, a small number of follow-up questions, mostly unrelated to the subject of this paper, will be asked. For example, the system will propose its best guess as to the morphological variants of nouns, verbs, and other words for the user to confirm or correct.

6. Intermediate Customizations

Having learned about each physical relation, TELI asks for information which, though not needed immediately, is either (a) more simply obtained at the outset, in a context relevant to its semantics, than at a later, arbitrary point, or (b) acquirable collectively, thus preventing several subsequent acquisitions. Unlike the initial acquisitions described in Section 5, intermediate customizations could be excised from the system without any loss in processing ability. We now summarize three forms of intermediate customizations, the last of which may be requested by the user at any time. Allowing users to ask for the other forms as well would be a simple matter.

First, the system will ask which columns contain values that either correspond to or are themselves English modifiers. In Figure 2-a, the values "1" through "G" in the "class" column might correspond (respectively) to "freshman" through "graduate student", in which case acquisitions might continue as

suggested in Figure 3. From this information, the system constructs a definition for each user-defined modifier; for example the internal definition of "sophomore" will be

((sophomore noun student) ((class p-noun) = 2))

A second intermediate acquisition, carried out subject to user confirmation, involves the acceptability of hypothesized syntax and semantics for (a) phrases based on "of", (b) phrases built around "have", "with", and "in", and (c) noun-noun phrases. In deciding what case frames to propose, TELI considers the information it has already acquired about simple functional ("of") relationships.

A third form of intermediate acquisition involves the system's invitation for the user to give lexical and syntactic information for one or more user-defined categories, namely titles, adjectives, common nouns, noun modifiers, prepositions, and verbs. For example, the user might specify six adjectives and the entities they modify, followed by four or five verbs and their associated case frames, and so forth.

7. On-Line Customization

In general, definitions are supplied to TELI whenever (a) an undefined modifier is encountered during the processing of an English input, or (b) the user asks to supply or modify a definition. In each case, the same methods are available for making definitions, and are independent of the modifier type being defined. When creating or modifying a meaning, users are presented with information as shown in Figure 4-a; upon asking to "add a constraint", they are given the menu shown in Figure 4-b. Multiple "constraints" appearring in a semantic specification are presently presumed to be *conjoined*.

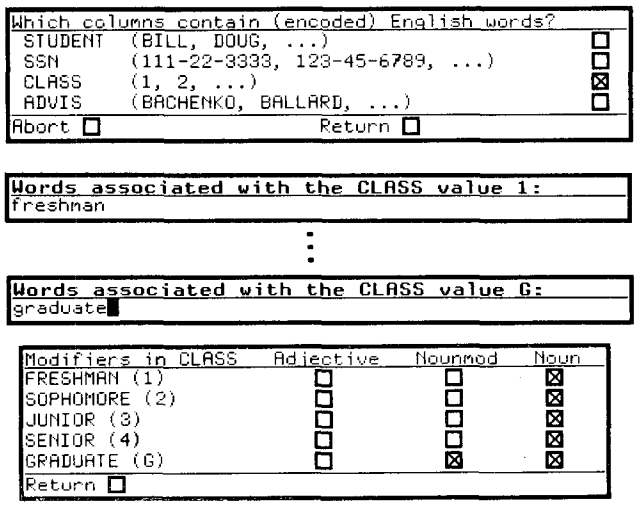

Figure 3: Intermediate Acquisitions

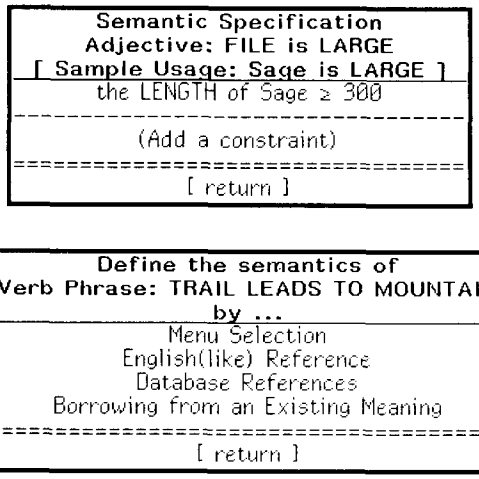

Figure 4: Top-Level Semantics Menus

As suggested in Figure 4-a and below, definitions are made in terms of *sample values*, which the system treats as formal parameters. In this way we avoid the problem of defining a phrase two or more of whose case slots may be filled by the same type of entity (cf. "a student is a classmate of a student if ..."). To assure that any domain value may appear as a constant, the user is able to alter the system's choice of sample names at any time.

7.1 Specification at the Database Level

As noted in Section 3, semantic specifications at the database level are primitive but useful. As shown in Figure 5, a database level specification comprises (a) a relation, possibly arrived at via a user-defined join, and (b) references to columns that correspond to the parameters of the phrase whose semantics is being defined. In many cases, the system can utilize its column type information, acquired as described in Section 5, to predict both the relation to be used (or pair of relations for joining) and the appropriate columns to join over, in which case the menu(s) that are presented will contain boldface selections for the user to confirm or alter.

7.2 Specification by Menu

In our previous experience with LDC, we found that a large variety of meanings could be defined by a predicate in which the result of some *function* is compared using some *relational operator* to a specified *benchmark* value. In TELI, we provide an enhancement to this scheme where definitions (a) may involve more than one argument, (b) may contain more than one function reference, and (c) are acquired in menu form. The current internal representation of a menu specification is a triple of the form suggested by

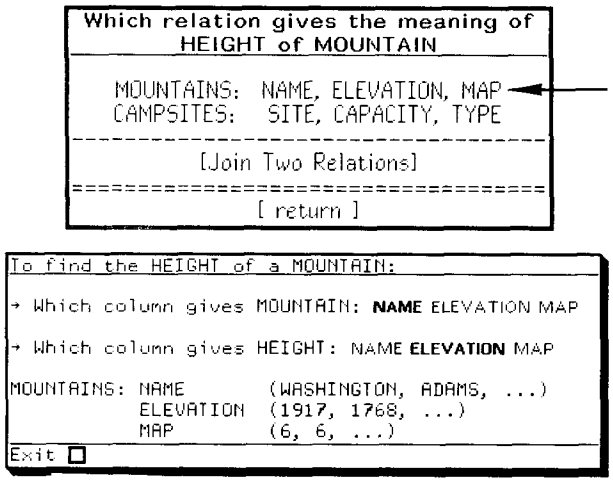

Figure 5: Database Specification

```
<spec>   -->  <term>  <relop>  <term>
<term>   -->  <atom>  |  <func> ( <atom> )
<atom>   -->  <constant>  |  <parameter>
<relop>  -->  =  |  <  |  <=  |  >  |  >=  |  ~=
```

An example of how menu semantics operates is given in Figure 6. When a semantics menu first appears, its "Function" field contains a list of all functions known to apply to at least one of the entities that the definition relates to. This reduces the number of keystrokes required from the user and, more importantly, helps guard against an inadvertent proliferation of concept names.

7.3 English and English-Like Specifications

In addition to the database and menu schemes just described, users may supply definitions in terms of English already known to the system. Some advantages to this are that (1) definitions may be arbitrarily complex, limited only by the coverage of the underlying syntactic component, and (2) users will implicitly be learning to supply semantics at the same time they learn to use the NLP itself. Some disadvantages are (1) a user might want to define something that cannot be paraphrased within the bounds of the grammatical coverage of the system, and (2) unless optimizations are carried out, references to user-defined concepts may entail inefficient processing.

An alternative to English specification, which functions similarly from the user's standpoint, is to provide for "English-like" specifications in which an expression supplied by the user is translated by some pattern-matching algorithm different from, and probably less sophisticated than, the process involved in actual English parsing. The primary advantage of English-like specification, over English specification, is that translations into internal form can be more efficient, since definitions or parts of definitions will be handled on a case by case basis. One probable disadvantage is that the scheme will be less general, in terms of definable concetps, and perhaps "spotty" in terms of what it makes available.

In TELI, both English and English-like specification are done in terms of sample domain values, which are treated as formal parameters. An example appears in Figure 7. In the current implementation, English-like specifications include (a) any definition definable by menu, and (b) definitions that involve (possibly negated) adjective or noun references. As of this writing, only English specifications that involve no nested parameter references can be processed.

7.4 Specification by Borrowing

In addition to whatever mechanisms an NL system specifically provides for semantic acquisitions, it is reasonable to allow users to define one meaning *directly* in terms of another (in addition to *indirect* dependence, as in the case of English specification). In TELI, users may ask to "borrow" from an existing meaning at any time. As shown in Figure 8, the system responds by finding all current items defined in terms of all or some of the parameters (i.e. entities) of the item for which the borrowing is being done. This assures that the entire borrowed meaning can be modified to apply to the item being defined. After being copied, a borrowed meaning may be edited just as though it had been entered from scratch.

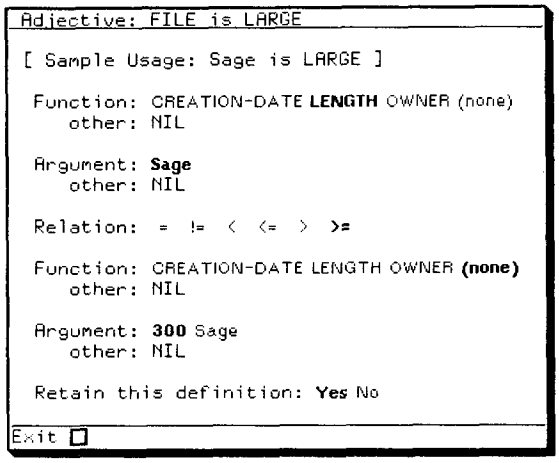

Figure 6: Menu Specification

Figure 7: English-like Specification

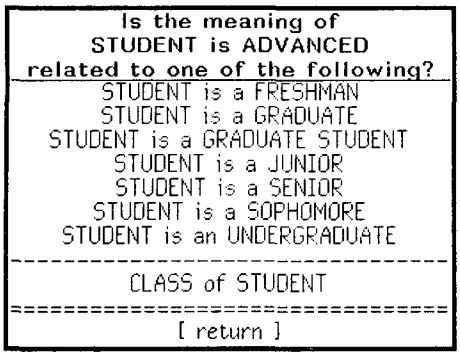

Figure 8: Borrowing a Meaning

8. Relation to Similarly Motivated Systems

At the most abstract level, our approach to transportability is unusual in that we have begun by building a *moderately sophisticated* NLP which, from the outset, *fundamentally* includes *replete customization facilities*. This contrasts with other efforts which have first built, perhaps over a period of several years, a *highly sophisticated system*, then sought to incorporate some customization features. Our work is also distinctive, though perhaps less so, in seeking to allow for customization by *end users*, as opposed to (say) a database administrator (cf. Thompson and Thompson, 1975, 1983, 1985; Johnson, 1985).

Some of the systems which, like **TELI,** seek to provide for user customization within the context of database query are **ASK** (Thompson and Thompson 1983, 1985), formerly REL (Thompson and Thompson, 1975), from Caltech; **INTELLECT,** formerly Robot (Harris, 1977), marketed by Artificial Intelligence Corporation; **IRUS** (Bates and Bobrow, 1983; Bates, Moser, and Stallard 1984), from BBN Laboratories; **TQA** (Damerau, 1985), formerly REQUEST (Plath, 1976), from IBM Yorktown Heights; **TEAM** (Grosz, 1983; Grosz et al, 1985), from SRI International; and **USL** (Lehmann, 1978), from IBM Heidleberg. Other high-quality domain-independent systems include **DATALOG** (Hafner and Godden, 1985), from General Motors Research Labs; **HAM-ANS** (Wahlster, 1984), from the University of Hamburg; and **PHLIQA** (Bronnenberg et al, 1978-1979), from Philips Research.

We now provide a comparison of TELI's customization strategies with those of the TEAM, IRUS, TQA, and ASK systems (other comparisons would also have been instructive, time and space permitting). Although we have recently spoken with at least one designer of each of these systems (see the Acknowledgements), it is possible that, in addition to intended simplifications, we may have overlooked or misunderstood certain significant, perhaps undocumented, features, in which case we apologize to the reader. Also, we note that our remarks are principally concerned with the *goals* and the *approaches* of various projects, and should not be viewed as commenting on the *accomplishments* or overall *quality* of TELI or any other system.

8.1 A Comparison with TEAM

Both TEAM and TELI represent English-language interfaces that have been applied to several moderately complex relational database domains. Each system provides for a variety of customizations by non-natural language experts, though neither system has claimed success with actual users in either customization or English processing mode. In terms of method, each system obtains (among other things) information about each column of each relation (table) of the database. We proceed to point out some of the more significant differences between the projects, as suggested by Grosz et al (1985) and indicated by Martin (1986).

To begin with, TEAM incorporates a more powerful natural language processor than does TELI, with provisions for quantifiers, simple pronouns, elaborate comparative forms, limited forms of conjunction, and numerous smaller features. Its "sort hierarchy" provides a taxonomy more general than that of TELI. It also incorporates disambiguation heuristics which seek to obviate the need for users to provide definitions for some phrase types (e.g. prepositional phrases based on "on", "from", "with", and "in"), and its preparations to deal with time and place references are without counterpart in TELI.

On the other hand, the customization features of TELI appear to offer greater sophistication, and sometimes more power, than the respective customization features of TEAM. In terms of sophistication, TELI always offers multiple ways of acquiring information, provides the ability to examine and borrow existing definitions, and is able to invoke the appropriate knowledge acquisition module when missing lexical, syntactic, or semantic information is required.

Copncerning definitional power, TELI generally provides for more complex definitions of words and phrases than does TEAM, as described in Sections 5-7. For example, whereas the SRI system typically requires a verb to map into some explicit or virtual relation (e.g. a join of explicit relations), TELI also allows an arbitrary number of properties of objects to be used in definitions (e.g. an *old* employee is one hired before 1980, or an employee *admires* a manager that works more hours than she does).

In TEAM, "acquisition is centered around the relations and fields in the database". In contrast, TELI provides several customization modes, as described in Section 3, and discourages low-level database specifications.

In contrast to the principles we espoused for TELI in Section 3, TEAM couples its methods of acquisition with the type of modifier being defined. For example, when seeing a "feature field", which contains exactly two distinct values, the system asks for "positive adjectives" and "negative adjectives" associated with these values (e.g. "volcanic" is a positive adjective associated with the database value "Y"). In TELI, these relationships arise as a special case of the acquisitions shown in Figures 3, 6, and 7b.

An interesting similarity between TEAM and TELI is that each provides for English(like) definitions. For example, TEAM might be told that "a volcano erupts", from which it infers that a mountain erupts just in case it is a volcano.

8.2 A Comparison with IRUS

Another recently developed facilitiy to allow user customizations of a database front-end is represented by the IRACQ component of the IRUS system (Ayuso and Weischedel, 1986). In addition to its practical value, IRACQ is intended as a vehicle that permits experimental work with sophisticated knowledge representation formalisms.

IRACQ is similar to TELI in shielding the user from the layout of the underlying data files. Another similarity is that each system accepts case frame specifications in English-like form, but IRACQ allows proper nouns as well as common nouns to be used. Thus, a user might suggest the case frame of the verb "write" by saying "Jones wrote some articles". Since IRUS provides for quite general taxonomic relationships among defined concepts (e.g. nouns), IRACQ proceeds to ascertain which of the possibly several classes that "Jones" belongs to is the most general one that can act as the subject of "write".

One important difference between TELI and IRACQ is that IRUS distinguishes *conceptual* information, which resides within its KR framework, from the *linguistic* information that characterizes the English to be used. Thus, while IRACQ supports definitions in terms of an arbitrary number of predicates, as does TELI, it assumes that any concepts needed to define a new language item have already been specified. These representations, acquired by a separate module called KREME, involve the KL-ONE notions of "concept" and "relation", which are similar to, but more sophisticated than, the 1- and 2-place predicates that come into existence during a session with TELI.

At present, IRACQ allows users to define *case frame* information for verb phrases, prepositional phrases, and noun phrases involving "of". Its treatment of prepositional phrases is very much like that of TELI in that the head noun being modified is considered part of the the noun-preposition-noun triple for which a definition is being acquired (cf. Section 4.1). Definitions for individual words (e.g. nouns and adjectives) are not supported but are being considered for future versions of the system, as are facilities that enable the system to inform the user of existing predicates that might be useful in defining a new language item. This facility will be similar in spirit to TELI's provisions for "borrowing" definitions, as described in Section 7.4.

8.3 A Comparison with TQA

Unlike most efforts at transportability, TQA has been designed as a *working prototype*, capable of being customizated for complex database applications by actual users. The primary responsibility of the customization module is to acquire information that relates *language* concepts, e.g. subject of a given verb, to the *columns* of the database at hand.

Like TELI, TQA avoids having to copy all database values into the lexicon by constructing "shape" information to recognize numbers and similar patterns. For example, the system might deduce that all database values referring to a department are of the form "letter followed by two digits", which allows for valuable disambiguations during parsing. Thus, in a database where employees manage projects and supervisors manage departments, the question "Who manages K34?" can be understood to be asking about supervisors without having to find "K34" in either the lexicon or the database.

A related problem, which TQA addresses more squarely than most systems (including TELI), concerns the appearance and possible equivalence of database values. For example, "vac lnd" might indicate "vacant land", "grn" and "green" might be used interchangeable, and so forth. Many practical applications require that these sorts of issues be addressed in order for a user to obtain reliable information.

Another useful feature concerns the acquisition of information that enables non-trivial output formatting. In simple cases, a database administrator might want nine-digit values appearing in columns associated with social security numbers to be printed with dashes at the appropriate points (e.g. 123456789 becomes 123-45-6789). In more complicated situations, values might actually need to be decoded, so that 0910 becomes "vacant land". This provision for decoding is similar to to the form of intermediate acquisition shown in Figure 3, though here it is being used for opposite effect.

8.4 A Comparison with ASK

The current ASK prototypes, which run on Sun, Vax, and HP desktop systems, are derived from earlier work on the REL system, which itself derives from work on the DEACON project, which stems from the early 1960's. Unlike most recent efforts, which have sought to incorporate customization features into an existing more-or-less single-domain system, the work with REL, the "Rapidly Extensible Language", fundamentally included definitional capabilities as early as 1969.

To begin with, ASK provides quite general customization facilities, allowing English definitions at least as sophisiticated as those outlined in Section 7.3. An example is "ships 'carry' coal to Oslo if there is a shipment whose carrier is ships, type is coal and destination is Oslo". Arithmetic facilities are also provided, e.g. "area equals length times beam".

The most distinguishing features of ASK, however, derive from the designers' desire to incorporate natural language technology into an intergrated information management system, rather than provide simple sentence-by-sentence database retrieval. One feature allows ASK to be connected to several *external database* systems, drawing information from each of them in the context of answering a user's question. A second feature allows a user to provide *bulk data input*. This begins with the interactive specification of a record type, followed by information used to populate the newly created relation.

Acknowledgements

The current TELI system derives from work on the LDC project, which was carried out at Duke University by John Lusth and Nancy Tinkham. In converting the NL portions of LDC to operate in our present context, we have engaged in frequent discussions with several persons, including Joan Bachenko, Alan Biermann, Marcia Derr, George Heidorn, Mark Jones, and Mitch Marcus. We also wish to thank Paul Martin of SRI, Damaris Ayuso and Ralph Weischedel of BBN, Fred Damerau of IBM Yorktown Heights, and Fred Thompson of Caltech, for their willingness to answer a number of questions that helped us to formulate the comparisons given in Section 8. Finally, we wish to thank Marcia Derr for many useful comments on a draft of our paper.

References

Ayuso, D. and Weischedel, R. Personal Communication, April 1986.

Ballard, B. "A 'Domain Class' Approach to Transportable Natural Language Processing", *Cognition and Brain Theory* 5, 3 (1982), 269-287.

Ballard, B. "The Syntax and Semantics of User-Defined Modifiers in a Transportable Natural Language Processor", *Proc. Coling-84*, Stanford University, July 1984, 52-56.

Ballard, B. "User Specification of Syntactic Case Frames in TELI, A Transportable, User-Customized Natural Language Processor", *Proc. Coling-86*, Bonn, West Germany, August, 1986.

Ballard, B., Lusth, J., and Tinkham, N. "LDC-1: A Transportable Natural Language Processor for Office Environments", *ACM Transactions on Office Information Systems* 2, 1 (1984), 1-23.

Ballard, B. and Tinkham, N. "A Phrase-Structured Grammatical Framework for Transportable Natural Language Processing", *Computational Linguistics* 10, 2 (1984), 81-96.

Bates, M. and Bobrow, R. "A Transportable Natural Language Interface for Information Retrieval", *Proc. 6th Int. ACM SIGIR Conference*, Washington, D.C., June 1983.

Bates, M., Moser, M. and Stallard, D. "The IRUS Transportable Natural Language Interface", *Proc. First Int. Workshop on Expert Database Systems*, Kiawah Island, October 1984, 258-274.

Bronnenberg, W., Landsbergen, S., Scha, R., Schoenmakers, W. and van Utteren, E. "PHLIQA-1, a Question-Answering System for Data-Base Consultation in Natural English", *Philips tech. Rev.* 38 (1978-79), 229-239 and 269-284.

Damerau, F. "Problems and Some Solutions in Customization of Natural Language Database Front Ends", *ACM Transactions on Office Information Systems* 3, 2 (1985), 165-184.

Grosz, B. "TEAM: A Transportable Natural Language Interface System", *Conf. on Applied Natural Language Processing*, Santa Monica, 1983, 39-45.

Grosz, B., Appelt, D., Martin, P. and Pereira, F. "TEAM: An Experiment In The Design Of Transportable Natural-Langauge Interfaces", *Artificial Intelligence*, in press.

Hafner, C. and Godden, C. "Portability of Syntax and Semantics in Datalog". *ACM Transactions on Office Information Systems* 3, 2 (1985), 141-164.

Harris, L. "User-Oriented Database Query with the ROBOT Natural Language System", *Int. Journal of Man-Machine Studies* 9 (1977), 697-713.

Johnson, T. *Natural Language Computing: The Commercial Applications*. Ovum Ltd, London, 1985.

Lehmann, H. "Interpretation of natural language in an information system", *IBM J. Res. Dev.* 22, 5 (1978), pp. 560-571.

Martin, P. Personal communication, March 1986.

Tennant, H. "Experience With the Evaluation of Natural Language Question Answerers", *Int. J. Conf. on Artificial Intelligence*, 1979, pp. 275-281.

Thompson, F. and Thompson, B. "Practical Natural Language Processing: The REL System as Prototype", In *Advances in Computers*, Vol. 3, M. Rubinoff and M. Yovits, Eds., Academic Press, 1975.

Thompson, B. and Thompson, F. "Introducing ASK: A Simple Knowledgeable System", *Conf. on Applied Natural Language Processing*, Santa Monica, 1983, 17-24.

Thompson, B. and Thompson, F. "ASK Is Transportable in Half a Dozen Ways", *ACM Trans. on Office Information Systems* 3, 2 (1985), 185-203.

Wahlster, W. "User Models in Dialog Systems", Invited talk at *Coling-84*, Stanford University, July 1984.