

User Specification of Syntactic Case Frames in TELI, A Transportable, User-Customized Natural Language Processor

Bruce W. Ballard

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

1. Introduction

In this paper, we present methods that allow the users of a natural language processor (NLP) to define, inspect, and modify any case frame information associated with the words and phrases known to the system. An implementation of this work forms a critical part of the Transportable English-Language Interface (TELI) system. However, our techniques have enabled customization capabilities largely independent of the specific NLP for which information is being acquired.

The primary goal of the syntactic acquisitions of TELI is to redress the fact that many NL prototypes have failed (1) to make known to users exactly what inputs are allowed (e.g. what words and phrases are defined) and (2) to meet the needs of a given user or group of users (e.g. appropriate vocabulary, syntax, and semantics). 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, 1977). Thus, we have chosen to make TELI "transportable" in an extreme sense, where customizations may be performed (1) *by end users*, as opposed to computer professionals, and (2) *at any time* during English processing.

The current prototype of TELI, which runs on a Symbolics Lisp Machine, derives from work at Duke University on the LDC project (Ballard, 1982; Ballard, 1984; Ballard, Luth, and Tinkham, 1984; Ballard and Tinkham, 1984). The top-level menu of TELI, and also a sample snapshot of a session with TELI, which may give the flavor of how the system operates, are shown in Figure 1. A discussion of semantic acquisitions appeared in Ballard and Stumberger (1986).

2. The Importance of Case Frame Information

Following Ballard and Tinkham (1984), TELI seeks to enable domain-independent English processing by maintaining *detailed* case frame information about the phrase types provided for by the system. For example, when accessing a restaurant database, the system would know not only that "serve" is a transitive verb but also that it requires objects of type Restaurant as subject and either Food or Meal as object. Thus, if

"Japanese" is known to be a type of Food, and "lunch" a Meal, then the system would accept

"Which restaurants serve lunch?"
"How many restaurants serve Japanese food?"

and reject

* "Which meals serve Japanese food?"
* "How many meals serve a restaurant?"

As a more elaborate example, suppose we are accessing information about researchers at Bell Labs, and we ask

"Which manager does the newest speech employee not in building two report to?"

The exact phrase types involved in the above input are

Verb Phrase: (employee report nil nil to manager)
Prepositional Phrase: (employee in building)
Noun-Noun Phrase: (project employee)

where "nil" denotes unfilled optional slots for direct object and particle. Adverbials ("not") and inflections of single-word modifiers ("newest") are handled by mechanisms separate from those associated with what we are calling phrase types (see Section 9).

As suggested above, we treat the noun being modified by a prepositional phrase as an *argument* of the modifier in question (e.g. "in"). Thus, departing from more conventional treatments, our "head noun" is part of the prepositional case frame, which therefore comprises three rather than two slots. Similarly, adjective phrase case frames comprise four rather than three slots (see Section 5). Our syntactic and semantic treatment of prepositional phrases is suggested by the "Intermediate Representation" shown in Figure 1.

3. Situations In Which Case Frames May Be Considered

There are presently five situations in TELI where users are able to examine and possibly modify syntactic case frames. The first of these occurs during initial customization, when the system first confronts a new

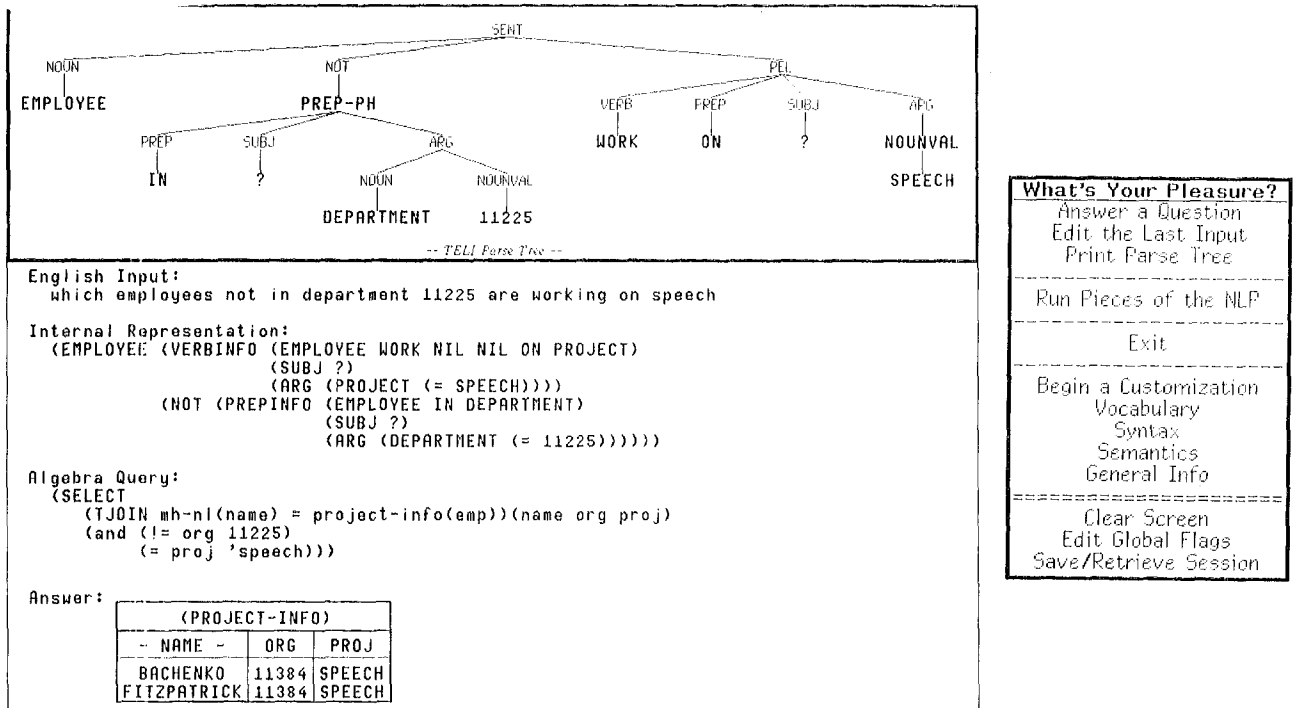


Figure 1: Sample Display Screen; Top-Level Menu of TELI

database. The remaining four, which concern the central English processing mode of TELI relevant to this paper, are as follows.

When explicitly requested by the user. When logging on to the system, or at a later time, users can ask what words and phrases are associated with a particular domain object. This contributes to the habitability of the interface.

When adding new vocabulary items. For example, if the user specifies "open" as an adjective, the system asks for its associated case frames.

When attempting to recover from a parsing failure. For example, if a sentence that the system cannot parse contains the word "with", the system will offer to show the user all existing prepositional triples of the form Entity-with-Entity. This permits the user to ascertain whether the parsing failure was caused by missing case frame information or for some other reason. If the problem is due to missing case frame information, the user can add it, then have the system retry the input.

When semantic information is being considered. Users may ask to examine or modify current definitions of prepositional phrases, verb phrases that take "up" as a particle, and so forth. To do so, the user first specifies the syntactic relationships of the phrases of interest.

As shown below, the manner in which the user specifies a phrase or range of phrases is independent of the reason the case frame information is being sought.

4. Principles Behind Case Frame Specifications

The primary criteria our methods of case frame specification are designed to meet are:

To be independent of the specific NLP that information is being supplied to. This permits us to alter or augment the underlying case frames used by the parser without having to change any of the actual code responsible for acquiring phrasal compatibilities from the user. For instance, we have made several changes in the way relational nouns like "classmate" are processed, without any changes to the customization modules.

To be fully data-driven. Our knowledge acquisition modules provide general capabilities for a large class of phrase types, but they know nothing specific about verb phrases, etc. This is our principal method of achieving the previous criterion. At present, about two pages of specifications of a form shown below are used to drive TELI's knowledge acquisition component.

To be driven by data which can in principal be inferred from the underlying grammar. This means that any changes to the grammar will be automatically reflected in the modules that acquire case frame information. At present, about half of the two pages of data that drive our syntactic knowledge acquisition module are taken directly from the grammar.

In addition to the above criteria, which relate to automating the process of customizing an NLP, an additional human factors criterion is to have types of

information which *function* similarly, from the *user's* standpoint, *appear* similar as presented by the system. Thus, output formats do not always reflect how information is stored and manipulated by the system.

5. Defining the User Interface to Case Frames

At present, TELI provides for five phrase types:

Adjective Phrase:

e.g. researchers *associated with* TELI

Noun-Modifier Phrase:

e.g. the *COLING* presentations

Verb Phrase:

e.g. employees *working with* Brachman

Prepositional Phrase:

e.g. the researchers *in* Marcus' department

Relational Noun Phrase:

e.g. the *associates* of Litman,
the *salary* of Smith

In discussing how actual case frame acquisitions are done, we will find it convenient to give in detail all the information associated with one of the phrase types provided for by the system grammar. For this purpose, we have chosen to consider *adjective phrases*, since the situations they involve are fairly representative. The actual system provides somewhat more sophisticated capabilities than what we have space to describe here, especially in its treatment of verb phrases.

Before proceeding, we note that the actual data structures used in TELI differ slightly from those presented here, although they contain precisely the same information. Also, we mention that our use of the term "interface designer" reflects our belief that most of the job about to be described can be done by a trained user of the system, as opposed to the actual system builders.

Before TELI is supplied with phrase type information, it will have been given lexical information about each part of speech recognized by the underlying grammar. Parts of speech are also classified as either "open" or "closed", the former enabling the user to supply new words of that type. For example, the system designer might have specified

Open = (adjective, noun, verb, ...)

Closed = (article, prep, ...)

This information is used by the acquisition module in deciding which case frame slots may be filled with vocabulary items not already in the system lexicon.

As a first step in telling the system about phrase types, the interface designer must indicate for each case frame slot (1) a *name* to be used inside the system to identify this slot, (2) an appropriate *filler* type, and (3) an *external name* to be used as a label in system output. For adjective phrases this might be given as

(adjinfo		
(head	entity	"Subject")
(adj	adjective	"Adjective")
(prep	prep	"Preposition")
(obj	entity	"Object")

where "adjinfo" is an arbitrary symbol used internally to reference adjective phrase case frames. *Slot names* (head, adj, prep, obj) are arbitrary; *filler types* (entity, adj, prep) generally correspond to parts of speech, although "entity" denotes the subset of nouns that comprise the primitive object types of the domain at hand. For example, in a building domain, Room might be a basic object type (entity), while "office" is merely a noun that refers to some of the objects of type Room. Finally, *external names* ("Subject" etc.) may be any string useful in identifying a case role.

Next, the interface designer specifies an arbitrary number of templates which the system will seek to match against a user's English-like case frame specification. For example,

(adjinfo (a Head can be Adj Prep an Obj))

enables the system to recognize a specification such as

"a room can be adjacent to a corridor"

as a reference to an adjective phrase case frame. Recall that this information is given by the interface designer and does not define, but merely reflects, the grammatical coverage provided by the underlying parser. Note that case frame templates are specified in terms of case frame labels rather than parts of speech. This allows transposing the elements of a case frame containing two or more elements of the same type.

In the event that the interface designer wishes to specify optional items, (s)he can either give multiple specifications or denote optionality within parentheses. Thus, the verb phrase specification

(subj verb (obj) (part) (prep obj))

will expand into eight patterns having from two to six elements each.

Since the interface designer will have specified slot names for each type of case frame, the system can easily detect the presence of "noise" words. In addition, small matters such as the equivalence of "a" and "an" must be taken care of, and the interface designer does this by giving a translation map such as

Noise-Translations = ((an a) (the a))

which instructs the system to make the indicated replacements in both an English-like specification to be matched and the internal patterns. It is not necessary that all noise words be present in the system dictionary. For example, "can" does not presently appear in the context of our question-answering applications.

6. A User's View of Case Frame Specification

There are two ways in which a user may designate which case frame information is of interest, namely (1) by *menu*, and (2) by *English-like specification*. The former is straightforward, while the latter is more convenient, and more interesting.

In specification by *menu*, the user first indicates a phrase type to be inquired about, and is then instructed by the system to provide a filler for each slot in the associated case frames. For example, to find out what domain objects can be "in" a county, a user would make the selections indicated in Figure 2. Since our case frames allow both the head-noun and argument-of-preposition slots to be filled with any basic object type of the domain at hand, the second and fourth menus contain the same options. The internal list that results from these specifications is essentially

Phrase Type:	Prepositional Phrase
Head Noun:	(all)
Preposition:	in
Object:	county

As suggested in Figure 2, during menu specification, the system considers in turn each case slot of the phrase type in question and, for each of them, presents to the user for selection a list of current fillers, along with an option to "look at all". For slots whose filler type is either an open category, or a closed category having possible fillers not presently being listed, an option to select some "other" filler is included. Finally, for optional phrase elements (e.g. direct object of a verb), an option appears that allows the user to select "none".

In *English-like* specification, the user types a phrase that indicates each desired slot value, not necessarily in the order they appear internally. Appropriate noise words may appear, and "?" may be used as a "wildcard" to indicate an interest in all possible values. For example, the sample specification given above by menu could be indicated by

"a ? can be in a county"

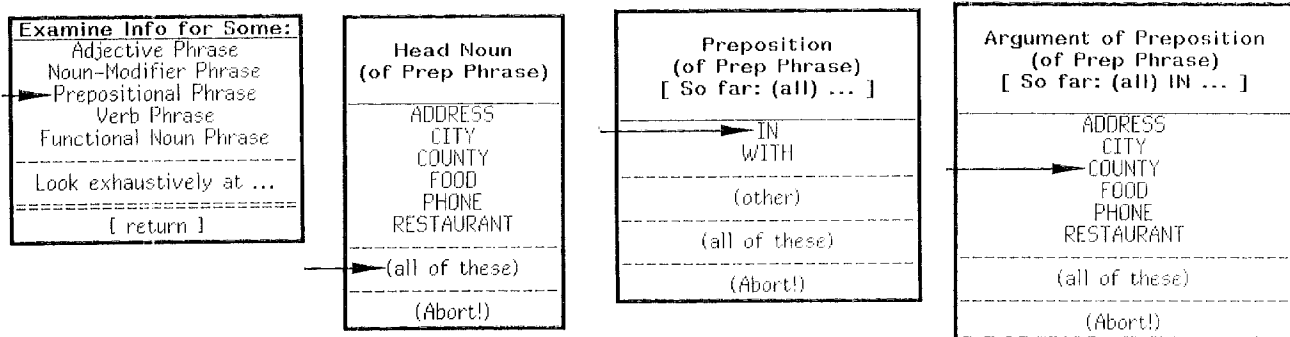


Figure 2: Menu Requests to See What Objects Can Be "in" a "county"

As with menu specification, it is possible for the user to introduce new vocabulary. For instance, if the italicized items were new in the specifications

"an employee can *report* to a manager"
 "an employee can be *responsible* for a project"
 "an employee can be the *supervisor* of a project"

the system would have sufficient information to find a unique match among the patterns stored. In these situations, the system will have automatically determined the part of speech of the new word.

Although our use of "?" may seem artificial in the example above, when compared against a more fluent method of inquiry such as "what can be in a county", it allows *any* case frame slot to be inspected, not just those slots that are filled with nominals. For example, a user might specify

"a city can be ? a county"

to find all prepositions linking "city" with "county", or

"an employee can ? a project ?"

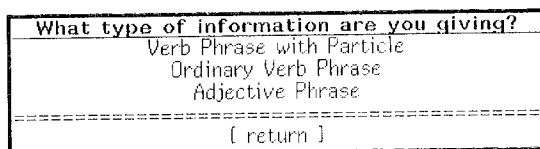
to find all verb-particle pairs connecting "employee" with "project". We prefer to provide a small number of simple and powerful mechanisms, even though other methods might appear preferable in some situations. For the reader whose aesthetics differ from ours, we note that alternate phrasings can be provided for by simple modifications to the algorithm given in Section 7.

Unlike menu specification, English-like specification allows certain ambiguities to arise, especially when the system designer has chosen to permit terse forms with few or no noise words. For example, the respective absence of the noise words "can" and "can be" in the specifications

"employee *responsible* for project"
 "employee *report* to manager"

makes it impossible for the system to decide whether the new word is an adjective or a verb. In such situations,

the system constructs a suitable menu, which for the above specifications would be roughly



In the other extreme, it is possible that none of the stored patterns match the user's specification, in which case the system requires the user either to paraphrase or to resort to menu specification.

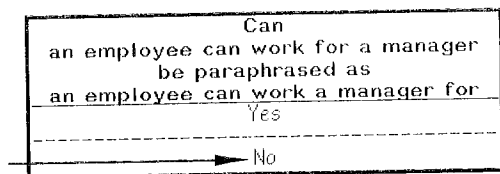
In our experience, English-like specification yields a unique match about 80 percent of the time; more than one match about 15 percent of the time; and no matches about 5 percent of the time. The most frequent situation in which a multiple match occurs concerns the possibility that a preposition appearing in a verb phrase is a particle. For example, if the user types

"an employee can pick a project up"

then "up" is known to be a particle by its position. If instead the user were to type

"an employee can pick up a project"

then the system will need to determine whether "up" is a particle. Although we generally avoid yes-no questions, as discussed below, we decided to allow one in this frequent and predictable situation, as indicated by



Finally, it is useful to allow the system to present the user with relevant information that the system knows it will need, rather than wait (and hope) for the user to offer it. As a first example, suppose the system has failed to parse the input

"Which corridor is Stumberger's office adjacent to?"

and the user accepts the system's offer to provide help in tracking down the problem. Since the word "adjacent" is an adjective, and adjectives are known to have phrases associated with them, the system will supply all current information about those adjective phrases having "adjacent" in the adj slot and leaving the remaining slots unspecified. That is, the system will respond as though the user has specified

Phrase Type:	Adjective Phrase
Head Noun:	(all)
Adjective:	adjacent
Preposition:	(all)
Object:	(all)

7. How English-Like Specifications are Processed

When an English-Like Specification is received from the user, the system must (1) determine what phrase type is being dealt with; (2) detect any new words; and (3) account for any unspecified (wildcard) case slots. As an example, suppose a user wants to know what things can be "associated with" an employee, and suppose further that the word "associated" is not yet known to the system. In this case, the system will naturally know of nothing that can be "associated with" an employee, but will give the user an opportunity to add to its knowledge. If the user were to type

an employee can be associated with a ?

this specification is first scanned and turned into

a employee can be ?? with a ?

where "??" marks the position of an unknown word and "?" continues to denote a wildcard slot. Note that (1) a "noise translation" from Section 5 has been used for "an", and (2) the noise words "can" and "be" have not yet been removed, since they may act as content words in a pattern for something other than an adjective phrase.

The next step is to substitute part-of-speech labels for each word in the partially processed specification. Only those parts of speech that the system knows are relevant, as indicated by the information supplied by the interface designer as shown above, are included (e.g. "a" is not replaced by "article"). Thus, the system converts the structure shown above into

a (noun entity) can be ?? (prep) a ?

at which point an attempt can be made to match the internal patterns that represent the acceptable case frame specifications.

The pattern matching that occurs at this point is simple, where

?	matches	any case slot
??	matches	any "open" category case slot
x	matches	x
(x y ...)	matches	any of x, y, ...

In particular, the single match found for the structure shown above is

a entity can be adj prep a entity

which is known to be associated with adjective phrases (since it was defined for that purpose).

At this point, the intermediate structure containing the ?? marker is re-examined and compared with the original specification the user typed; the user is asked to confirm that "associated" is indeed a new adjective; and the lexicon update routine is invoked to insert "associated" into the lexicon as an adjective.

Next, the system strips noise words and so the case frames to be examined are indicated by

Phrase Type: Adjective Phrase
 Head Noun: employee
 Adjective: associated
 Preposition: with
 Object: (all)

Finally, the system presents a 1-dimensional menu, similar to that shown below in Figure 3b, which allows the user to specify what things an employee can be associated with.

8. Display of Relevant Information

The formats that we have chosen for TELI to display the current case frame information relevant to a user's specification are based on the desires

1. to allow information to be *inspected* and *updated* simultaneously, and
2. to *minimize* the number of *specific menu types* presented to the user.

In particular, the system constructs, whenever possible, a menu in which each possible setting of unspecified case frame values may independently be turned "on" or "off" by a mouse click. In the current implementation, "whenever possible" amounts to precisely those situations in which no more than two case frame slots are left unspecified. Thus, a menu will contain choice boxes which have from zero through two dimensions, according to the number of unspecified case slots. Examples appear in Figures 3a through 3c. Note that appropriate row and column labels, and also a suitable menu label, must be constructed by the system. Since the system has no initial domain-specific vocabulary, these menus must be formulated at run-time.

When more than two case slots are unspecified, the system simply prints all existing case frames that satisfy the indicated constraints, supplying an initially filled box for each, as indicated in Figure 3d. This allows the user to remove individual case frames, and the "Add" option allows information to be added.

Although we have chosen to avoid asking literal yes-no questions whenever possible, largely because of the low information content they provide, the choice-box scheme we have adopted implicitly asks a number of simultaneous yes-no questions. Thus, when the user checks the box in a menu for the preposition "with" having City as row label and County as column label, (s)he is in effect answering "yes" to the implicit question "can a city be in a county".

9. Discussion

We now consider (a) treatment of single-word modifiers, (b) planned enhancements to case frame capabilities, and (c) related acquisition modules.

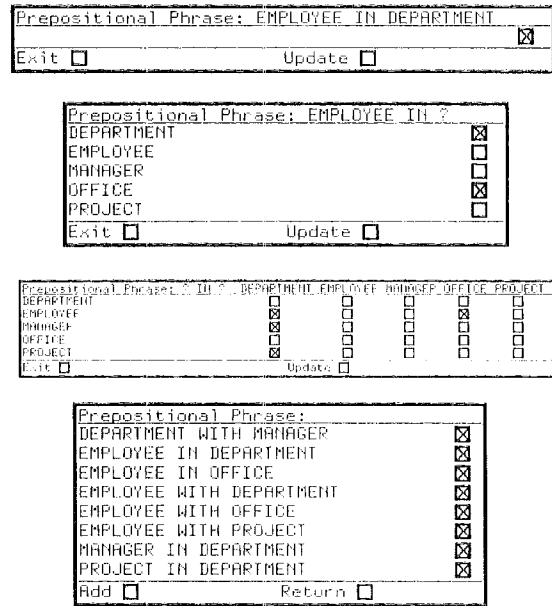


Figure 3: Case Frame Display for Varying Number of Queried Slots

The techniques presented in this paper, which are directed toward case frames for multiple-word phrases, are actually used for single-word modifiers as well. Internally, one important difference is that the associated modifier compatibility information is maintained in the lexicon rather than stored into auxiliary case frames. As an example, if the user says that the word "large" can modify objects of type Department and Office, one associated lexical entry is

(larger compar large (nt department office))

As with case frames, the user may impart compatibility information for single-word modifiers by either menu or English-like specification. The latter is typified by

a room can be large

while an example of how the user may ask to see everything known about acceptable adjective modifications is shown in Figure 4.

Several enhancements to our facilities for English-like capabilities are planned. For instance, we noted in section 6 that whereas the use of "?" to denote an unspecified slot works for all parts of speech, it might be more natural to denote unspecified *nouns* by



Figure 4: Modifier Information for Adjectives

"what" and possibly transpose the specification accordingly. As noted previously, the question is one of generality versus naturalness in specific situations; simple modifications to the algorithm given in the preceding section would enable alternate forms. We are considering whether to alter our methods of inquiry, perhaps to provide for both forms. Another enhancement being considered is to permit inflected forms, as indicated by the italicized elements of

students can be failed by an instructor

Finally, we wish to give some feeling for the lexical and semantic acquisition facilities alluded to in the paper. Figure 4a gives the top-level menu pertaining to part-of-speech information. This menu enables the user to obtain output which as with case frame information allows simultaneous inspection and modification, as illustrated in Figure 4c. Word and phrase meanings are acquired similarly, and also involve either menu or English-like specification. As an example of the latter, if the user has said that

an employee can work with an employee

then the system will ask what "work with" means in this sense by selecting two example employees in terms of which the user is asked to define semantics. For example, the system will in effect ask

What does it mean for Bob to work with Jill?

at which point the user might say

the dept of Bob is equal to the dept of Jill

10. Related Work

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 (Martin et al, 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 (Hoepfner et al, 1983; Wahlster, 1984), from the University of Hamburg; and PHLIQA (Bronnenberg et al, 1978-1979), from Philips Research.

Due to the space limitations endemic to conference papers, we refer the reader to Ballard and Stumberger (1986) for some substantive comparisons, largely related to semantic issues, between TELI and each of TEAM, IRUS, TQA, and ASK.

References

- 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., 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 Stumberger, D. Semantic Acquisition in TELI: A Transportable, User-Customized Natural Language Processor. *Proc. 24th Annual Meeting of the ACL*, Columbia University, June 1986.
- 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. *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.
- 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., Appelt, D., Martin, P. and Pereira, F. TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces. *Artificial Intelligence*, to appear.
- 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.
- Hoepfner, W., Christaller, T., Marburger, H., Morik, D., Nebel, B., O'Leary, M., and Wahlster, W. Beyond Domain-Independence: Experience with the Development of a German Language Access System to Highly Diverse Background Systems. *Proc. 8th Int. J. Conf. on Artificial Intelligence*, Karlsruhe, West Germany, 1983, pp. 588-594.
- 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., Appelt, D. and Pereira, F. Transportability and Generality in a Natural Language Interface System. *Proc. 8th Int. J. Conf. on Artificial Intelligence*, Karlsruhe, West Germany, 1983, pp. 573-581.
- Moser, M. Domain Independent Semantic Acquisition. BBN Report No. 5694, Research in Knowledge Representation for Natural Language Understanding, September 1984.
- 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.