A Knowledge Framework for Natural Language Analysis

Paul S. Jacobs
AI Branch, GE Corporate R & D
Schenectady, NY 12301 USA

Abstract

Recent research in language analysis and language generation has highlighted the role of knowledge representation in both processes. Certain knowledge representation foundations, such as structured inheritance networks and feature-based linguistic representations, have proved useful in a variety of language processing tasks. Augmentations to this common framework, however, are required to handle particular issues, such as the ROLE RELATIONSHIP problem: the task of determining how roles, or slots, of a given frame, are filled based on knowledge about other roles. Three knowledge structures are discussed that address this problem. The semantic interpreter of an analyzer called TRUMP (TRansportable Understanding Mechanism Package) uses these structures to determine the fillers of roles effectively without requiring excessive specialized information about each frame.

I. Introduction

A variety of work in language analysis [Bobrow and Webber, 1980, Sondheimer *et al.*, 1984, Lytinen, 1984, Hirst, 1987] and language generation [Jacobs, 1985, Sondbeimer and Nebel, 1986] exploits some STRUCTURED knowledge representation. A structured representation is one in which entities, or FRAMES, and their slots, or ROLES, may be related to more abstract objects and roles, as in KL-ONE [Brachman and Schmolze, 1985], FRAIL [Charniak *et al.*, 1983], or KODIAK [Wilensky, 1986]. The main advantage of structured representations for natural language is that knowledge about how a given frame or role is expressed linguistically may be taken as default knowledge about the expression of more specific frames and roles.

SEMANTIC INTERPRETATION is the part of the language analysis process that consists of constructing a correct, complete representation of the content of a natural language input. The task is difficult because there are no hard and fast rules about the relationship between linguistic structure and underlying meaning. For example, consider the following inputs:

- (Ia) FVank was sent the message by Jones.
- (Ib) The message was sent FVank by Jones.
- (Ic) The job was sent to the line printer by Jones.
- (2a) How many arguments does the command take?
- **(2b)** 'Rm' gives you the names of your files.

Sentences **(1a-c)** illustrate distinct senses of the verb "send", even in the computer domain, as well as the problem of determining the role of Frank in the British English of **(1b)**, structurally identical to **(1a)**. **(2a)** and **(2b)** demonstrate metaphorical senses of "take" and "give", metaphorical because a command operating upon arguments is not actively *taking* them, and printing a list of files is not really *giving* because you don't receive anything. Practically, we want to avoid representing the senses as *giving* or *taking* because the other concepts involved violate constraints or SELECTIONAL RESTRICTIONS on the roles they would play: Commands are not suited to being *givers* or *takers* as they are not animate. A variety of work in linguistics and AI [Chafe, 1968, Becker, 1975, Wilks, 1978] has argued that metaphors such as these may be the rule as much as the exception.

A major problem in interpreting sentences such as the above is the following:

- *The Role Relationship Problem:* The problem of determining, for a given candidate frame, what the conceptual fillers are of the roles of that frame, given a combination of abstract and specialized linguistic knowledge. In **(1a-c)** candidate frames are *sending-electronic-mail* and *transmission-to-printing-device;* in **(2a)** candidates include *command-execution* and *command-producing-output.* The semantic interpreter must determine, for example, that Frank fills the *source* role and Jones the *addressee* of the *sending-electronic-mail* event in **(1a)** or that the arguments mentioned in **(2b)** fill the *input* role of the command. This problem is important because it should not be necessary to specify, for each potential frame, the extensive knowledge required to fill the roles of that frame appropriately.

This paper describes how the features of a representation called ACE apply to the problem described above. The representation is used both by the KING generator [Jacobs, 1985) and a language analyzer, TRUMP (TRansportable Understanding Mechanism Package) [Jacobs, 1986], in a variety of applications.

II. The Ace Framework

Ace is a set of tools originally designed to promote extensibility by encoding knowledge about language in a uniform manner [Jacobs and Rau, 1985]. As with some of the work with KL-ONE [Bobrow and Webber, 1980, Sondheimer *et al,* 1984], the idea of Ace is to use the same structured representation, KODIAK [Wilensky, 1986], for both linguistic and conceptual structures, but the representation is augmented by having explicit structures that tie together linguistic and conceptual roles.

The discussion that follows considers the Ace knowledge structures used in representing these role relationships and how these structures can be used to determine appropriate role fillers in linguistic processing.

A. Linguistic Relations

Linguistic relations in Ace represent the linguistic structures to which conceptual roles are associated. These relations are organized hierarchically, so that often the syntactic realization of a linguistic relation derives from a more abstract category, while its meaning is represented at a more specific level.

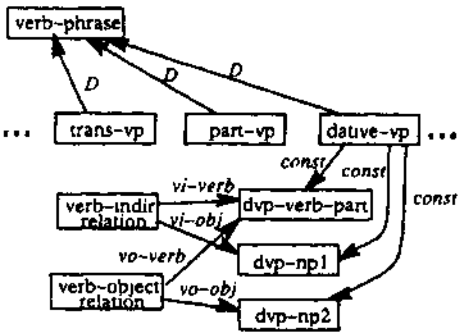

Figure 1: Linguistic relations and syntactic structures

Figure 1 shows the Ace representation of two linguistic relations applicable to the examples discussed here. The "D" link in these figures stands for *DOMINATE,* the KODIAK term for subsumption, or "a-k-o". Common relations such as *verb-indir-relation* and *verb-object-relahon* are associated with syntactic structures; since these relations might appear in a range of surface forms, the association of linguistic and conceptual information to the relations adds flexibility to the system.

B. The REF Association

Linguistic relations provide a hierarchy of linguistic roles that are associated with conceptual interpretations. The explicit link between linguistic and conceptual structures is accomplished using REF. Having explicit knowledge about linguistic expression makes it easier to use the same knowledge for both analysis and generation and promotes the interaction between abstract and more specialized linguistic knowledge. For example, the relationship between the indirect object and the conceptual *recipient* is a high-level REF association that may combine with particular knowledge about verbs of a more concrete nature.

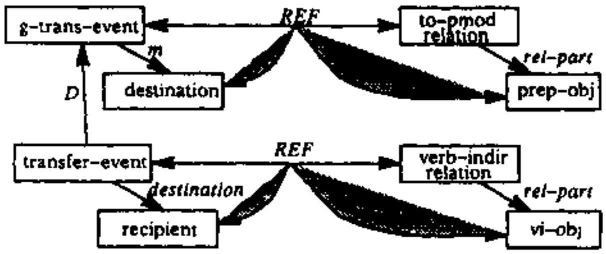

Figure 2: REF links linguistic and conceptual structures

Two important REF relationships in Ace are illustrated in Figure 2. The association of the *to-pmod-relation* at the level of *g-trans-event* for "generalized transfer events", a higher level than the *verb-indir-relatwn,* represents the knowledge that the object of the preposition "to" describes a more general role, *destination,* than the *recipient* described by the indirect object of a verb. Generalized transfer events include physical transfers as well, while in transfer events the recipient must abstractly receive the transferred object.

REF addresses the role relationship problem by providing a general means of associating linguistic and conceptual roles. Unlike systems such as Hirst's [Hirst, 1987], such relationships are often derivable from the knowledge structures in the network rather than being required explicitly for each word/sense template. For example, the relationship between the indirect object of the verb "sell" and the *customer* role is derived from the ROLE-PLAY between *indirect-object* and *recipient* and the fact that the *customer* plays a *recipient* role in the transfer of merchandise (cf. [Jacobs, 1985]). With the preposition "to", as in "sold the book to Fred", the role of *customer* is filled by virtue of its ROLE-PLAY relationship with the *destination* role.

C. The VIEW Association

The VIEW association is similar to REF, except that it represents metaphorical relationships rather than associations between linguistic and conceptual roles. For exampie, there is a broad class of English "transfer" expressions such as "give a kiss", "take a punch", and "get a back rub". The generalization here may be captured as a conceptual relationship, namely *being acted upon is often viewed as a transfer, in which the patient of the activity is viewed as the recipient.*

The example of "taking arguments" is similar, in that commands do not really "give" or "take" or "get" anything, but the language is extremely consistent in describ-

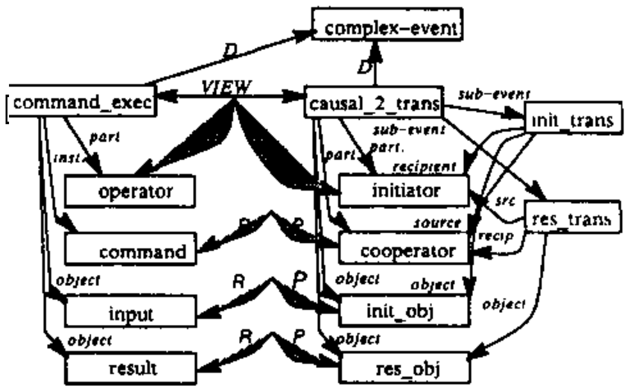

Figure 3: Capturing metaphorical generalizations

ing them as if they did. Figure 3 shows a VIEW relationship used to handle expressions like "getting arguments", "taking arguments", and "giving back a result". This representation has been slightly abbreviated due to its complexity; in reality the common manner of describing the complex *command-execution* event belongs to a broader class of metaphors along with "The frosting takes two cups of sugar". The VIEW as illustrated, however, encompasses quite a range of expressions. The power of this VIEW comes from the ROLE-PLAYs that capture the consistency of the metaphorical expressions, such as that relating *command* to *cooperator* and thereby to the *source* role of an initial transfer in an exchange (*causaL2.trans).* VIEW thus provides a tool for handling numerous role relationships that otherwise would be enumerated as special cases in the system.

JJJ. The TRUMP Analyzer

TRUMP (TRansportable Understanding Mechanism Package) [Jacobs, 1986] is a natural language system designed for use in a variety of domains. The theoretical basis of the system is that the Ace knowledge base design makes it possible for a core of linguistic knowledge to be exploited across domains.

The algorithm that TRUMP uses to perform linguistic analysis consists of the following mechanisms: (1) A syntactic parser, which identifies linguistic constituents and instantiates linguistic relations that are tied to matched structures, (2) A mapping mechanism, which produces conceptual structures by following VIEW and REF links from these instantiated linguistic structures, and (3) A phase called *concretion,* which finds the most specific frame suggested by other information and fills out the roles of that frame. For more details of this process, the reader is referred to [Jacobs, 1986].

Figure 4 is a sketch of the operation of the TRUMP

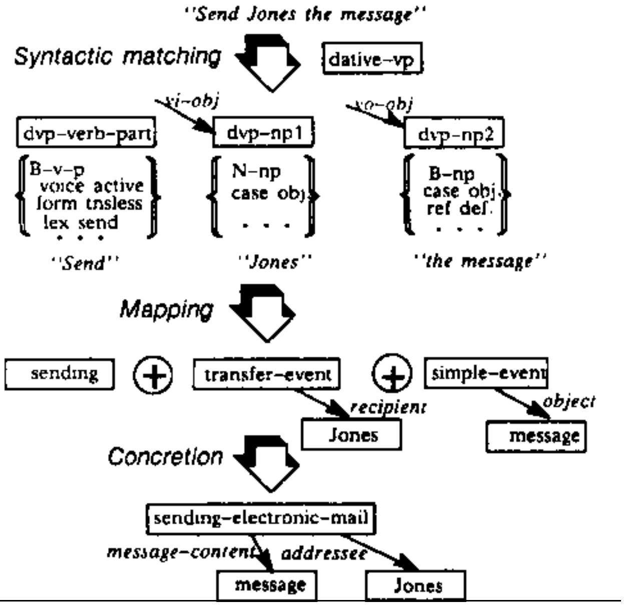

Figure 4: Operation of TRUMP

system, showing the results of the syntactic, mapping, and concretion processes. The application of role relationships is a major part of the work done by the semantic interpreter, or phases (2) and (3) above. When a linguistic structure is mapped, the interpreter must find the role in the resulting concept that appropriately corresponds to the linguistic role. Often this involves following a chain of ROLE-PLAY associations, and occasionally it does not completely specify the resulting role. For example, in "John gave Mary a kiss", versus "John gave Mary's cheek a kiss," the semantic interpreter correctly determines that the linguistic indirect object may express either the surface being kissed or the person being kissed. The semantic constraints on these roles favor the more likely interpretation,

TRUMP currently consists of about 5,000 lines of Common Lisp code, including the parser, semantic interpreter, knowledge base manipulation, and lexical acquisition mechanisms. The "core" Ace knowledge base consists of several hundred linguistic entries with about a thousand concepts, but this is not large enough to select a new domain without coding a fair amount of new knowledge, The system is currently being used, experimentally, in four drastically different natural language projects. While we are far from having a program that can easily function within a new application area, the exercise is proving useful in identifying natural language tools that do apply across domains.

IV. Relation to other Systems

Three knowledge-based systems are especially similar in design to TRUMP. These are the KL-ONE based work described by Sondheimer, Weischedel and Bobrow [Sondheimer *et al,* 1984], the Absity program of Hirst [Hirst, 1987], and Lytinen's MOPTRANS [Lytinen, 1984].

The KL-ONE based work is closest in knowledge base design to TRUMP, as the systems use similar, uniform representational frameworks for both the linguistic "syntaxonomy" and the conceptual knowledge base. The translation rules of this system, corresponding closely to the REF associations of Ace, can also be applied through inheritance, thus facilitating the use of generalized role relationships. These rules, however, are not expressed in a declarative form; thus it is not as clear how they would apply to language generation, and it proves difficult to handle metaphorical expressions.

Hirst's system also applies a similar knowledge organization, but for theoretical reasons Hirst does not allow role relationships to be inherited in his system, requiring case-role specifications for each word sense. The "Polaroid words" in Hirst's system correspond closely to the concretion mechanism of TRUMP, in that roles are filled and more specific frames activated as more information propagates from the linguistic mechanism of the system.

Lytinen's MOPTRANS system is similar to TRUMP not only in the organization of knowledge structures (MOPs) but also in the choice of some of the higher level frames that are used to derive specialized interpretations via frame selection. MOPs, however, do not really support structured inheritance: although constraints and prototypes may be assigned at the concept/role level, a role or slot of a particular frame cannot derive from multiple roles. Thus, as in Absity, much of the knowledge used to handle role relationships must be handled at a very specific level. Also, this knowledge is attached to the individual MOPs, rather than being declaratively represented.

V. Summary and Conclusion

The ROLE RELATIONSHIP problem in semantic interpretation is the task of appropriately filling out the roles or slots of a candidate frame. Three knowledge structures of Ace—LINGUISTIC RELATIONS, REF ASSOCIATIONS, and VIEW ASSOCIATIONS—help to alleviate this problem by providing a hierarchy of linguistic structures and explicit role relationships that include metaphorical expressions. These structures are particularly useful in choosing among candidate concepts and in appropriately filling the roles of selected concepts. They combine with a representation such as KODIAK or KL-ONE to form an enriched framework for language processing. The TRUMP language analysis system supports the application of this core knowledge about language across domains.

References

[Becker, 1975] J. Becker. The phrasal lexicon. In *Theoretical Issues in Natural Language Processing,* Cambridge, Massachusetts, 1975.

[Bobrow and Webber, 1980] R. Bobrow and B. Webber. Knowledge representation for syntactic/semantic processing. In *Proceedings of the National Conference on Artificial Intelligence,* Palo Alto, California, 1980.

[Brachman and Schmolze, 1985] R. Brachman and J. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science,* 9(2), 1985.

[Chafe, 1968] W. L. Chafe. Idiomaticity as an anomaly in the Chomskyan paradigm, *foundations of language,* 6(1), 1968.

[Charniak *et al,* 1983] E. Charniak, M. Gavin, and J. Hendler. *The FRAIL/NASL reference manual* Technical Report CS-83-06, Brown University Department of Computer Science, 1983.

[Hirst, 1987] G. Hirst. *Semantic Interpretation and the Resolution of Ambiguity.* Cambridge University Press, Cambridge, England, 1987.

[Jacobs, 1985] P. Jacobs. *A Knowledge-Based Approach to Language Production.* PhD thesis, University of California, Berkeley, 1985.

[Jacobs, 1986] P. Jacobs. Language analysis in not-so-limited domains. In *Proceedings of the Fall Joint Computer Conference,* Dallas, Texas, 1986.

[Jacobs and Rau, 1985] P. Jacobs and L. Rau. Ace: associating language with meaning. In T. O'Shea, editor, *Advances m Artificial Intelligence,* North Holland, Amsterdam, 1985.

[Lytinen, 1984] S. Lytinen. *The Organization of Knowledge in a Multi-lingual, Integrated Parser.* PhD thesis, Yale University, 1984.

[Sondheimer and Nebel, 1986] N. Sondheimer and B. Nebel. A logical form and knowledge-base design for natural language generation. In *Proceedings of the Fifth National Conference on Artificial Intelligence,* Philadelphia, 1986.

[Sondheimer *et al,* 1984] N. Sondheimer, R. Weischedel, and R. Bobrow. Semantic interpretation using KL-ONE. In *Proceedings of the Tenth International Conference on Computational Linguistics,* Palo Alto, 1984.

[Wilensky, 1986] R. Wilensky. Knowledge representation - a critique and a proposal. In J. Kolodner and C. Riesbeck, editors, *Experience, Memory, and Reasoning,* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1986.

[Wilks, 1978] Y. Wilks. Making preferences more active. *Artificial Intelligence,* 11, 1978.