Achieving Bidirectionality

Paul S. Jacobs
Artificial Intelligence Program
GE Research and Development Center
Schenectady, NY 12301 USA

Abstract

The topic of BIDIRECTIONALITY, using common knowledge in language processing for both analysis and generation, is of both practical and theoretical concern. Theoretically, it is important to determine what knowledge structures can be applied to both. Practically, it is important that a competent natural language system be able to generate outputs that are relevant to the inputs it understands, without excessive redundancy. This problem revolves around the ability to relate linguistic structures declaratively to their meaning.

1 Introduction

BIDIRECTIONALITY, or the ability to use a common knowledge base for both language analysis and generation, is a desirable feature of a real language processing system. A natural language "front end" must not only perform syntactic analysis, but must derive a suitable representation of a meaning or intention from a linguistic input. A natural language generator performs the inverse task of producing a linguistic utterance from a meaning or intention. A bidirectional system performs both tasks using as much shared knowledge as possible.

Two practical concerns motivate this work: (1) A system that uses shared knowledge for analysis and generation will produce output in the subset of language that it understands, thus avoiding inconsistences between the input and output, and (2) Using shared knowledge avoids the inefficiency of having distinct encodings of the same linguistic information.

The first concern, having a natural language interface "speak" the same language it understands, is more than a convenience. Responses in a dialogue often use a word or phrase that has been mentioned by another speaker. This cannot be done effectively unless the word or phrase is common to both the input and output language. A computer user will expect the system to understand a phrase or construct that the system has itself used; this aggravates the consequences of inconsistencies between input and output language. Moreover, if an interface is to be transportable across domains, a distinct subset of language will be applicable to each domain. The bidirectional knowledge base allows both the input and output to be constrained simultaneously.

The second concern, efficiency of knowledge representation, becomes more compelling as the lexical and semantic capabilities of natural language systems increase. While there is ample motivation for having a common grammar for analysis and generation, the need for a common lexicon is even stronger. Having two lexicons is counterintuitive; what makes practical sense is to have a single lexicon indexed differently for generation from analysis. Now that many systems have more and more knowledge built into their lexicons, the effects of redundancy become more drastic. When more information is required of the lexicon, however, the difficulties in developing a shared lexicon are more pronounced.

The principal concern in designing a natural language system that performs both analysis and generation, therefore, is a bidirectional lexicon. The main issue to be considered here is what information must be included in this lexicon and how bidirectional lexical knowledge should be structured.

2 Issues Regarding Bidirectionality

There has been very little research in language generation relative to language understanding and syntactic analysis. A negligible amount of research has addressed the problem of bidirectionality. Some work has touched on shared knowledge of lexical semantics [Jacobs, 1985, Steinacker and Buchberger, 1983] and on grammatical frameworks suitable for bidirectional systems [Kay, 1984]. At the recent TINLAP (Theoretical Issues in Natural Language Processing) conference [Wilks, 1987], position papers brought out a number of points concerning bidirectionality that had not previously appeared in the literature. The positions largely embraced the need for knowledge shared between analysis and generation while laying out the practical reasons why bidirectional systems are not prevalent.

A good summary of issues in bidirectionality is found in [Mann, 1987]. Each aspect of the generation process can be related to some part of language analysis that seems to draw from common knowledge. However, the processes themselves as well as the problems involved in building actual language processing systems differ, to such an extent that scientists do not find the time to attend to the common issues. Another point is that both fields, especially generation, largely ignore the problem of lexical semantics [Marcus, 1987], a problem that might help to bring the tasks closer together.

It is a mistake to treat analysis and generation as completely independent tasks. Given that the goal of much of natural language research is to program computers to communicate in the way people do, the ideal natural language program must use natural language as both a "front end" and a "back end". Knowledge that has historically been used more in generation, pertaining to text structure, coherence, and constraints on lexical choice, influences the analysis task. Knowledge primarily applicable to analysis, such as vocabulary and grammatical coverage, and information applied to ambiguity and vagueness, can be applied to generation as well. The problem of linguistic knowledge base design is thus fundamentally different for a bidirectional system.

3 The Bidirectional Lexicon

Several characteristics are essential to a lexicon that can be used effectively in both analysis and generation:

1. Principally, the lexicon and knowledge base of the system must be declarative; all the material must take the form of data structures rather than rules or program code.
2. The semantic component of the lexicon; i. e. the representation of word meanings and word senses, must be sufficient to guide lexical choice in generation and to resolve vague or ambiguous words in analysis.
3. Lexical collocations, phrasal lexemes, and grammatical constructions must be represented. This compound lexical knowledge is necessary in generation because the selection of a particular word influences the selection of other words in a phrase, even when the phrase is internally grammatical. The knowledge is important in analysis in so far as it can aid in handling multiple word senses.

Most systems satisfy the declarative requirement above, although the degree to which knowledge is proceduralized varies greatly from one model to another. The second and third requirements, the richness of lexical semantics and the need for compound knowledge, are more often overlooked. In generation, a lexical entry that lists a word stem and a corresponding set of linguistic and semantic features is not enough; what is needed is a relationship between the lexical item and a knowledge representation structure [Jacobs, 1986] and a means of selecting the lexical item from among the other possible words [Mathiessen, 1981]. A word choice is not made independently from other choices; lexical choices have a direct influence on other lexical choices [Jacobs, 1985].

Lexical knowledge used primarily for generation can impact the way language analysis is performed, and vice versa. The following simple examples help to illustrate how complex lexical knowledge required for generation can also affect understanding:

- (1a) Hit the "return" key.
- (1b) Hit the "x" key.
- (2a) Type "return".
- (2b) Type "x".
- (3a) Hit "return".
- (3b) Hit "x".

A generation system needs a variety of lexical knowledge in order to produce utterances such as (3a), which is natural for most native speakers. In addition to knowledge about the word sense of "hit", the system must know what keys are suitable for "hitting", as well as that "hit" is used to describe striking a single key. This detailed lexical knowledge should also avoid using (2a) in place of (3a), since one cannot use "type" for a key that does not produce a character or text. Now, given that this knowledge is required for the appropriate generation of the utterances above, it makes sense that it should be used in determining the difference in meaning between (2a) and (3a) (the former means "Hit the sequence of keys r-e-t-u-r-n). In designing a system strictly for analysis, one would tend to distinguish (2a) from (3a) by assuming "hit" to have a different meaning from "type", and thus produce two incorrect but relatively subtle effects: First, the meanings of (2b) and (3b) would also be different, and second, (3b) would be equally acceptable to (3a).

Because a generation system must have enough information in the lexicon to make appropriate lexical choices, it must have lexical knowledge that relates the specific word senses above to the linguistic context in which they are used. A linguistic analyzer can then use this knowledge to make more accurate interpretations of the same words. This is a typical way in which lexical choice and word sense determination are related.

4 FLUSH

An example of a lexicon designed with the three characteristics described in the previous section is FLUSH (Flexible Lexicon Using Structured Hierarchical knowledge) [Besemer and Jacobs, 1987]. FLUSH combines a hierarchical phrasal lexicon [Wilensky and Arens, 1980, Jacobs, 1985, Dyer and Zernik, 1986] with declarative relations between language and meaning [Jacobs and Rau, 1985]. For example, figure 1 shows part of the lexical knowledge about the preposition "to", used in a prepositional phrase modifying either a verb or noun. The lexical relation *to-pmod* represents this linguistic category, and constrains how it can be used in a surface structure, based on its membership in the more general *mod-rel* (modifying relation) category.

Figure 2 shows how the *to-pmod* relation is associated with a generalized transfer event (either a physical transfer or a transfer of possession), with the object of the preposition describing the destination of the transfer. The link marked "REF" in figure 2 represents this sort of association between a linguistic and a conceptual structure. More specific transfers, as well as metaphorical "VIEWs" of transfers, are also explicitly represented in this diagram. Knowledge about senses of "sell", "tell", and "send", as well as constructs using such verbs, is thus represented in a neutral fashion.

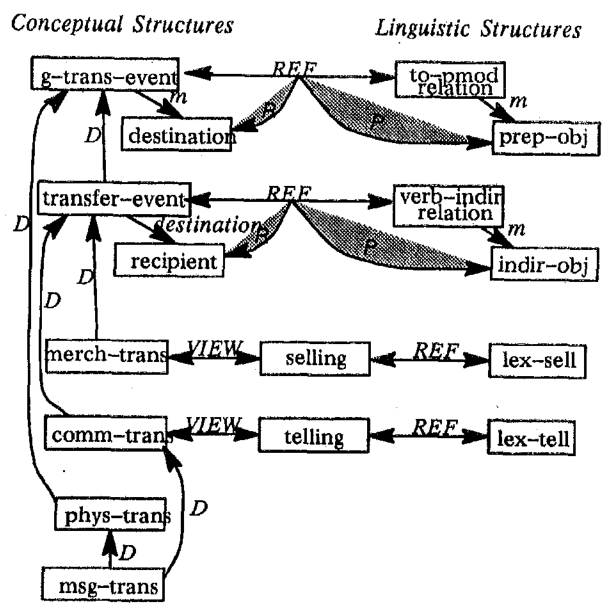

Figure 2: Relating linguistic and conceptual structures

Compound lexical knowledge, often involving figurative expressions, is also represented declaratively in FLUSH. Figure 3 shows how such knowledge is encoded: *lc-give-hug*, the lexical category for "giving a hug" and other variations on the same expression, belongs to a general category, *linguistic/conceptual*, which accounts for its linguistic flexibility such as its potential use in the passive voice. A "REF" association links *lc-give-hug* to the *hugging* concept, indicating declaratively that these expressions describe a *hugging* action rather

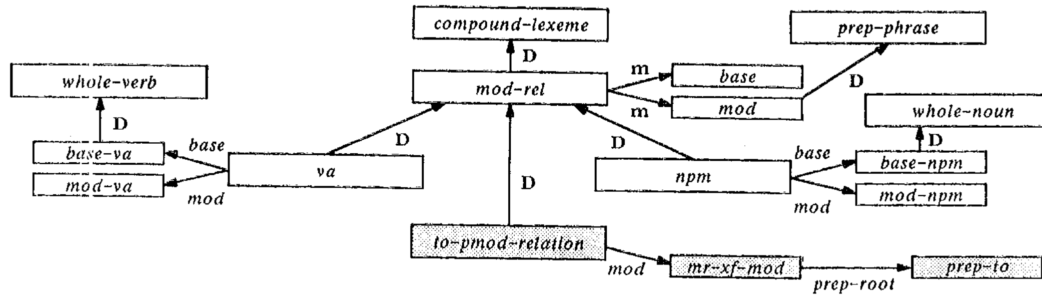

Figure 1: The modifying-relation compound-lexeme hierarchy.

than a literal sense of "give".

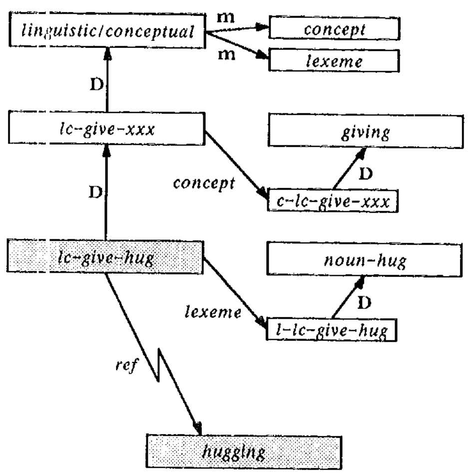

Figure 3: The linguistic/conceptual relation *lcr-give-hug*.

These examples, while only touching upon the lexical representation of FLUSH, shows some of the characteristics of a birectional lexicon. The hierarchy of linguistic structures allows access to these structures for both analysis and generation. Declarative links between linguistic and conceptual entities allow specific knowledge about linguistic expression to be used in both processes. The current task is to encode enough information in this form so that analysis and generation alike can be robustly performed.

5 Conclusion

Using certain knowledge for both analysis and generation is desirable in a natural language system, for both theoretical and practical reasons. This bidirectionality aids efficiency as well as insuring compatibility between analysis and generation components. A lexicon designed for bidirectionality differs distinctly from one designed for either generation or analysis alone, and often develops aspects of each process that might otherwise be overlooked.

References

[Besemer and Jacobs, 1987] David Besemer and Paul S. Jacobs. FLUSH: a flexible lexicon design. In *Proceedings of the 25th Meeting of the Association for Computational Linguistics*, Palo Alto, California, 1987.

[Dyer and Zernik, 1986] Michael G. Dyer and Uri Zernik. Encoding and acquiring meanings for figurative phrases. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, New York, 1986.

[Jacobs, 1985] Paul S. Jacobs. PHRED: a generator for natural language interfaces. *Computational Linguistics*, 11(4), 1985.

[Jacobs, 1986] Paul S. Jacobs. Knowledge structures for natural language generation. In *Proceedings of the Eleventh International Conference on Computational Linguistics*, Bonn, Germany, 1986.

[Jacobs and Rau, 1985] Paul S. Jacobs and Lisa F. Rau. Ace: associating language with meaning. In Tim O'Shea, editor, *Advances in Artificial Intelligence*, pages 295–304, North Holland, Amsterdam, 1985.

[Kay, 1984] M. Kay. Functional Unification Grammar: a formalism for machine translation. In *Proceedings of the Tenth International Conference on Computational Linguistics*, Palo Alto, California, 1984.

[Mann, 1987] W. C. Mann. What is special about natural language generation research? In *TINLAP-3: Theoretical Issues in Natural Language Processing-3*, pages 206–210, Computing Research Laboratory, New Mexico State University, Las Cruces, New Mexico, 1987.

[Marcus, 1987] M. Marcus. Generation systems should choose their words. In *TINLAP-3: Theoretical Issues in Natural Language Processing-3*, pages 211–214, Computing Research Laboratory, New Mexico State University, Las Cruces, New Mexico, 1987.

[Mathiessen, 1981] C. Mathiessen. A grammar and a lexicon for a text production system. In *Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics*, pages 49–56, Stanford, California, 1981.

[Steinacker and Buchberger, 1983] I. Steinacker and E. Buchberger. Relating syntax and semantics: the syntactico-semantic lexicon of the system VIE-LANG. In *Proceedings of the First European Meeting of the ACL*, Pisa, Italy, 1983.

[Wilensky and Arens, 1980] R. Wilensky and Y. Arens. PHRAN–A knowledge-based natural language understander. In *Proceedings of the 18th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, 1980.

[Wilks, 1987] Y. Wilks. *TINLAP-3: Theoretical Issues in Natural Language Processing-3*. Computing Research Laboratory, New Mexico State University, Las Cruces, New Mexico, 1987.