THE LEXICAL SEMANTICS OF COMPARATIVE EXPRESSIONS IN A MULTI-LEVEL SEMANTIC PROCESSOR

Duane E. Olawsky
Computer Science Dept.
University of Minnesota
4-192 EE/CSci Building
200 Union Street SE
Minneapolis, MN 55455
[olawsky@umn-cs.cs.umn.edu]

ABSTRACT

Comparative expressions (CEs) such as "bigger than" and "more oranges than" are highly ambiguous, and their meaning is context dependent. Thus, they pose problems for the semantic interpretation algorithms typically used in natural language database interfaces. We focus on the comparison attribute ambiguities that occur with CEs. To resolve these ambiguities our natural language interface interacts with the user, finding out which of the possible interpretations was intended. Our multi-level semantic processor facilitates this interaction by recognizing the occurrence of comparison attribute ambiguity and then calculating and presenting a list of candidate comparison attributes from which the user may choose.

1 PROBLEM DESCRIPTION.

Although there has been considerable work on the development of natural language database interfaces, many difficult language interpretation problems remain. One of these is the semantic interpretation of comparative expressions such as those shown in sentences (1) through (3).

(1) Does ACME construct **better** buildings than ACE?

(2) Does ACME construct buildings **faster** than ACE?

(3) Are **more oranges** than apples exported by Mexico?

To interpret a comparative expression (CE) a natural language processor must determine (1) the entities to be compared, and (2) the attribute(s) of those entities to consider in performing the comparison. The selection of comparison attributes is made difficult by the high level of lexical ambiguity exhibited by comparative predicates. For example, what pieces of data should be compared to answer query (1)? If the database contains information about foundation type, structural characteristics, wiring, and insulation, any of these attributes could be used. Similarly, when comparing orange and apple exports as in query (3), we might compare numeric quantity, weight, volume, or monetary value. To further complicate matters, the plausible comparison attributes for a comparative predicate change with the arguments to which that predicate is applied. Table 1 shows several examples of likely comparison attributes to use with the predicate "bigger" depending on the types of entity that are being compared. Since the system must determine for a comparative predicate the lexical definition intended by the user, this problem is, at heart, one of lexical ambiguity resolution.

The problems discussed so far are similar to the well known vagueness and context sensitivity of adjectives (although they occur here even in sentences without adjectives such as (3)). Any proposed method of CE interpretation should also treat several other phenomena that are unique to comparatives. These are bipredicational comparisons, cross-class comparisons, and pairability constraints. **Bipredicational comparisons** involve two predicates, as shown in example (4) (the

Table 1: Examples of argument sensitivity in the meaning of "bigger".

Argument type	Possible Comparison attributes
hotels	number of rooms
hospitals	number of beds
houses	square feet number of rooms, or number of bedrooms
wheat farms	number of acres
dairy farms	number of cows
countries	number of people, or land area
cars	length, curb weight, passenger space, or passenger limit

predicates are in boldface), and they use a different comparison attribute for each argument of the comparative.

(4) John's car is **wider** than Mary's car is **long**.

Bipredicational CEs have strong **pairability constraints** (Hale 1970). That is, there are restrictions on the pairing of predicates in a bipredicational CE. Example (5) gives a sentence that is semantically anomalous because it violates pairability constraints.

(5) ? Bob's car is **wider** than it is **heavy**.

A **cross-class** comparison involves arguments of radically different types as shown in (6).

(6) Is the Metrodome **bigger** than Ronald Reagan?[1]

Interpreting this comparison requires that we find a stadium attribute and a person attribute which are in some sense comparable (e.g. stadium-height and person-height). Pairability constraints also apply indirectly to cross-class comparisons as can be seen in the oddness of (7).

(7) ? The party was **longer** than my car.[2]

Although we have only one predicate ("longer") in this sentence, it is difficult to find a comparable pair of attributes. The attribute describing the length of a party is not comparable to any of the attributes describing the length of a car.

When faced with ambiguous input a natural language interface has two options. In the first one, it guesses at what the user wants and provides the answer corresponding to that guess. In the second, it interacts with the user to obtain a more completely specified query. Although Option 1 is easier to implement, it is also inflexible and can lead to miscommunication between the user and the interface. With Option 2, the system lets the user select the desired interpretation, resulting in greater flexibility and less chance of misunderstanding. It is the second option that we are exploring. To carry out Option 2 for CE interpretation the system must present to the user a list of the permissible comparison attribute pairs for the given CE. In Section 3 we will see how pairability constraints can be used to delimit these pairs. Comparatives add significant expressive power to an interface (Ballard 1988), and it is therefore important that reliable techniques be developed to resolve the lexical ambiguities that occur in CEs.

2 PRIOR WORK.

For purposes of discussion we will divide comparative expressions into the following commonly used classes: **adjectival**, **adverbial**, and **adnominal**, where the comparative element is based on an adjective, an adverb, or a noun, respectively. See (1)-(3) for an example of each type. Within linguistics, adjectival comparatives are the most studied of these three varieties. (See (Rusiecki 1985) for a detailed description of the various types of adjectival comparative.) For work on the syntax of CEs see (Bresnan 1973), (Pinkham 1985) and (Ryan 1983). Klein (1980), (1982) presents a formal semantics for adjectival CEs without using degrees or extents. It would be difficult to apply his work computationally since there is no easy way to determine the positive and negative extensions of adjectives upon which his theory rests. Hoeksema (1983) defines a set-theoretic

[1] Although this is an unusual comparison to request, it is perfectly understandable, and the literal interpretation is easily answered. As pointed out to me by Karen Ryan, sentence (6) has several possible metaphoric interpretations (e.g. "Does the Metrodome get more news coverage than Ronald Reagan?"). In this paper we will generally ignore metaphoric interpretations. However, using the approach we describe below, they could be handled in much the same way as the more literal ones.

[2] Sentence (7) can perhaps be interpreted metaphorically (perhaps with humorous intent), but it seems more difficult to do so than it does with (6). It is certainly hard to imagine what truth conditions (7) might have!

semantics for adjectival comparatives based on primitive grading relations that order the domain with respect to gradable adjectives. His primary concern is the relationship of comparatives to coordination and quantification, and he pays little attention to lexical ambiguities. Cresswell's work (Cresswell 1976) handles both adjectivals and adnominals and is closer in spirit to our own (see Section 3.1). It contains analogs of our Codomain Agreement Principle, mappings and base orders. The main difference is that whereas Cresswell always uses degrees, we also allow base orders to be defined directly on the domain entities.

Most of the work done on lexical ambiguity resolution (e.g. (Hirst 1984) and (Wilks 1975)) has focussed on homonymy (when words have a small number of unrelated meanings) rather than polysemy (when words have many closely related meanings) as occurs with CEs. The techniques developed for homonymy depend on large semantic differences between meanings and thus are not as useful for CEs.

Although comparatives are frequently used as examples in the NLP literature (e.g. (Hendrix, Sacerdoti, Sagalowicz, and Slocum 1978), (Martin, Appelt, and Pereira 1983) and (Pereira 1983)), no one has presented a detailed treatment of the ambiguities in the selection of comparison attributes. Most NLP researchers provide neither a detailed explanation of how they treat comparatives nor any characterization of the breadth of their treatment. Two exceptions are the recent papers of Ballard (1988) and Rayner and Banks (1988). The former treats adjectival and adnominal comparatives, and is primarily concerned with the interpretation of expressions like "at least 20 inches more than twice as long as". The selection of comparison attributes is not discussed in any detail. Rayner and Banks (1988) describe a logic programming approach to obtaining a parse and an initial logical formula for sentences containing a fairly broad range of CEs. They do not discuss lexical semantics and thus do not deal with comparison attribute selection.

This paper is an abbreviated version of a longer paper (Olawsky 1989), to which the reader is referred for a more detailed presentation.

3 SOLUTION APPROACH.

In this section we describe a rule-based semantic processor that follows Option 2. To provide for user-controlled comparison attribute selection we augment the common lexical translation process (e.g. (Bronnenberg, Bunt, Landsbergen, Scha, Schoenmakers, and van Utteren 1980) and (Ryan, Root, and Olawsky 1988)) with a Mapping Selector that communicates with the user and returns the results to the rule-based translator. The implementation of the approach described here is in progress and is proceeding well.

3.1 Semantic Description of Comparatives.

We base our approach on the semantic interpretation of a comparative predicate as a set-theoretic relation. A comparison defined by the relation $\mathcal{R}$ is true if the denotations of the first and second arguments of the comparative predicate (i.e. its **subject** and **object**[3]) form an element pair of $\mathcal{R}$. It is tempting to claim that comparatives should be defined by orders rather than relations (we call this the **Comparison Order Claim**). However, it can be shown (Olawsky 1989) that the comparison relation L_w for a bipredicational comparative like *longer than ... wide* is neither asymmetric nor antisymmetric[4], and hence, L_w is not an order.[5] Comparison relations are not defined directly in our semantic description. Instead they are specified in terms of three components: a **base order**, a **subject mapping**, and an **object mapping**. The base order is a set-theoretic order on some domain (e.g. the obvious order on physical lengths). The subject mapping is a mapping from the domain of the denotation of the subject of the CE to the domain of the base order (e.g. the mapping from a rectangle to its length). The object mapping is defined analogously. Let comparison relation $\mathcal{R}$ be defined by the base order B, and the subject and object mappings M_s and M_o. Then $\langle a, b\rangle \in \mathcal{R}$ if and only if $\langle M_s(a), M_o(b)\rangle \in B$. It should be noted here that comparison attribute selection is now recast as the selection of subject and object mappings.

[3] Our reasons for calling the first and second arguments of a CE the subject and object are syntactic and beyond the scope of this paper (see (Ryan 1983)).

[4] It is also easy to show that L_w is nontransitive.

[5] Klein ((1980), p. 23) and Hoeksema ((1983), pp. 410–411) both make claims similar (but not identical) to the Comparison Order Claim. It seems to us that bipredicationals pose a problem for Hoeksema's analysis (see (Olawsky 1989)). Klein appears to relax his assumptions slightly when he deals with them. Cresswell (1976) clearly avoids the Comparison Order Claim.

By definition, the subject and object mappings must have the same codomain, and this codomain must be the domain of the base order. We call this the **Codomain Agreement Principle**, and it is through this principle that pairability constraints are enforced. For example, when interpreting the CE in sentence (5), we must find a subject mapping for the width of Bob's car and an object mapping for its weight, and these mappings must have the same codomain. However, this is impossible since all width mappings will have LENGTH as a codomain, and all weight mappings will have WEIGHT as a codomain. The Codomain Agreement Principle also helps explain the interpretation of sentences (6) and (7).

Before concluding this section we consider the semantic description of CEs in TEAM ((Grosz, Haas, Hendrix, Hobbs, Martin, Moore, Robinson, and Rosenschein 1982) and (Martin, Appelt, and Pereira 1983)), comparing it to ours. Since comparative expressions were not the main focus in these papers, we must piece together TEAM's treatment of CEs from the examples that are given. In (Grosz, Haas, Hendrix, Hobbs, Martin, Moore, Robinson, and Rosenschein 1982), the CE "children older than 15 years" is translated to ((*MORE* OLD) child2 (YEAR 15)) where "*MORE* maps a predicate into a comparative along the scale corresponding to the predicate" (p. 11). This implies that TEAM requires the same mapping to be used for both the subject and object of the comparative. That would not work well for bipredicational CEs, and could also lead to problems for cross-class comparisons. In (Martin, Appelt, and Pereira 1983) the examples contain predicates (e.g. *salary-of* and *earn*) which, on the surface, are similar to mappings. However, in contrast to our approach, it does not appear that any special significance is given to these predicates. There is nothing in either paper to indicate that the many types of CEs are consistently translated to a base order, subject mapping and object mapping as is done in our system. Furthermore, there is nothing analogous to the Codomain Agreement Principle discussed in either paper. Now, we move on to a presentation of how the semantic description presented above is applied in our system.

3.2 General Comments.

We use a multi-level semantic processor (see (Bates and Bobrow 1983), (Bronnenberg, Bunt, Landsbergen, Scha, Schoenmakers, and van Uteren 1980), (Grosz, Haas, Hendrix, Hobbs, Martin, Moore, Robinson, and Rosenschein 1982), (Martin, Appelt, and Pereira 1983) and (Ryan, Root, and Olawsky 1988) for descriptions of similar systems). At each level queries are represented by logic-based formulas (see (Olawsky 1989) for examples) with generalized quantifiers ((Barwise and Cooper 1981), (Moore 1981) and (Pereira 1983)) using predicates defined for that level. The initial level is based on often ambiguous English-oriented predicates. At the other end is a description of the query in unambiguous database-oriented terms (i.e. the relation and attribute names used in the database). Between these levels we have a domain model level where formulas represent the query in terms of the basic entities, attributes and relationships of the subject domain described in a **domain model**. These basic concepts are treated as unambiguous. Linking these levels are a series of translators, each of which is responsible for handling a particular semantic interpretation task.

In this paper we restrict our attention to the translation from the English-oriented level (EL) to the domain model level (DML) since this is where CEs are disambiguated by choosing unambiguous mappings and base orders from the domain model. To perform its task the EL-DML translator uses three sources of information. First, it has access to the domain model, a frame-based representation of the subject domain. Second, it uses the semantic lexicon which tells how to map each EL predicate into a DML formula. Finally, this translator will, when necessary, invoke the Mapping Selector—a program that uses the semantic lexicon and the domain model to guide the user in the selection of a comparison attribute pair.

For our semantic formulas we extend the usual ontology of the predicate calculus with three new classes: sets, mass aggregations, and bunches. Sets are required for count noun adnominal comparatives (e.g. "Has ACME built **more warehouses** than ACE?") where we compare set cardinalities rather than entity attribute values. Given a class of mass entities (e.g. oil), a **mass aggregation** is the new instance of that class resulting from the combination of zero or more old instances. For example, if John combines the oil from three cans into a large vat, the oil in that vat is an aggregation of the oil in the cans. It is not necessary that the original instances be physically combined; it is sufficient merely to consider them together conceptually. Mass aggregations are needed for mass noun adnominal comparatives. Finally, we define the term **bunch** to refer ambiguously both to sets and to mass aggregations. Bunches are used in EL where mass aggregations and sets are not yet distinguished. Sets, mass aggregations and bunches are described in semantic formulas by the **SET-OF**, **MASS-OF**, and **BUNCH-OF** relations, respectively. These relations are unusual in that their second arguments are unary predicates serving as characteristic functions defining the components of the first argument—a set, aggregation or bunch. For example, *(*MASS-OF* m* $(\lambda w[(wheat\ w)]))$ is true in case m is the aggregation of all mass entities e such that $\lambda w[(wheat\ w)]\ (e)$ is true (i.e. e is wheat).

3.3 Base Orders and Mappings.

EL and DML formulas contain, for each CE, a base order and two mappings. Two sample EL base orders are *more* and *less*. DML base orders are typically defined on domains such as VOLUME, and INTEGER, but they can also be defined on domains that are not usually numerically quantified such as BUILDING-QUALITY, or CLEVERNESS. *More* and *less* are ambiguous between the more specific DML orders.

Most EL mappings μ correspond one-for-one with an English adjective (or adverb). They are binary relations where the first argument is an entity e from the domain and the second is the degree of μ-ness that e possesses. For example, if *big′* is an EL mapping, then in *(big′ e b)*, b is the degree of bigness for e. Of course, *big′* is ambiguous. In contrast to adjectival and adverbial CEs, all adnominals use the ambiguous EL mapping **MUCH-MANY** which pairs a bunch with its size.

In most cases, a DML mapping is a relation whose first argument is an entity from some class in the core of the domain model and whose second argument is from the domain of a base order. In the mapping predication *(DM_w-storage-volume w v)* the first argument is a warehouse, and the second is a volume. *DM_w-storage-volume* could serve as the translation of *big′* when applied to a warehouse. CEs based on count nouns generally use the **CARDINALITY** mapping which is like other mappings except that its first argument is a *set* of entities from a domain model class rather than a member of the class. The second argument is always an integer. Mass noun comparatives require a slightly different approach. Since we are dealing with a mass aggregation rather than a set, the **CARDINALITY** mapping is inapplicable. To measure the size of an aggregation we combine, according to some function, the attribute values (e.g. weight or volume) of the components of the aggregation.[6] Thus, the mappings used for mass adnominal comparatives are based on the attributes of the appropriate class of mass entities.

3.4 EL-DML Translation Rules.

As stated above, EL and DML are linked by a translator that uses rules defined in the semantic lexicon (see (Olawsky 1989) for sample rules). These rules constitute definitions of the EL predicates in terms of DML formulas. Our system employs three kinds of translation rules—Trans, MTrans, and BTrans. **Trans** rules have four components: a **template** to be matched against an EL predication, an **EL context specification**, a **DML context specification**, and the **DML translation** of the EL predication.[7] The context specifications are used to resolve ambiguities on the basis of other predications in the EL formula and the (incomplete) DML formula. A rule is applicable only if its context specifications are satisfied. Although a predication in an EL context specification must *unify* with some predication in the context, subsumption relationships are used in matching DML context specifications. Thus, the DML context specification *(DM_building b)* will be satisfied by *(DM_warehouse b)* since *DM_building* subsumes *DM_warehouse*. **MTrans** rules are intended for the translation of subject and object mapping predications from EL to DML. They have two extra components that indicate the base order and the mapping to be used in DML. This additional information is used to enforce the Codomain Agreement Principle and to help in the user interaction described in Section 3.5. Finally, **BTrans**

[6] Although the aggregation function would likely be SUM for attributes such as weight, volume, and value, other functions are possible. For example, AVERAGE might be used for a nutritional-quality attribute of an agricultural commodity. The aggregation function is not explicitly reflected in our system until the database level.

[7] Trans rules are nearly identical to the lexical translation rules used in the ATOZ system (Ryan, Root, and Olawsky 1988). However, our rules do have some additional features, one of which will be discussed below.

rules are used to translate *BUNCH-OF* predications to DML.

One noteworthy feature of our translation rules is that they can look inside a functional λ-argument to satisfy a context specification.[8] We call these **λ-context specifications**, and they may be used inside both EL and DML context specifications for rules of all three types. However, it is only in BTrans rules that they can occur as a top level specification. Top level λ-context specifications (e.g. *(λb [(DM_building b)])*) are matched to the functional argument of the relevant *BUNCH-OF* predication. This match is performed by treating the body of the λ-context specification as a new, independent context specification which must be satisfied by predications inside the body of the functional argument. In Trans and MTrans rules, a λ-context specification can occur only as an argument of some normal predicational context specification. For example, the specification *(*MASS-OF* b (λc [(DM_commodity c)]))* can be used in any DML context specification. It checks whether *b* is a mass of some commodity. Just as standard context specifications provide a way to examine the properties of the arguments of a predication being translated, λ-context specifications provide a way to determine the contents of a bunch by inspecting the definition of its characteristic function.

Before continuing, we compare our context matching mechanism to the similar one used in the PHLIQA1 system (Bronnenberg, Bunt, Landsbergen, Scha, Schoenmakers, and van Utteren 1980). This system uses a typed semantic language, and context checking is based entirely on the type system. As a result, PHLIQA1 can duplicate the effect of context specifications like *(DM_building b)* by requiring that *b* have type DM_building. However, PHLIQA1 cannot handle more complex specifications such as *((DM_building b) (DM_b-owner b ACME))* since there is no semantic type in PHLIQA1 that would correspond to this subset of the buildings in the domain.[9] The same comments apply to λ-context specifications which can be declared in PHLIQA1 by specifying a functional semantic type. That is, *(λb (DM_building b))* is written as the type DM_building → truthvalue, a function from buildings to truth values. As with standard context specifications, *(λb (DM_building b) (DM_b-owner b ACME))* cannot be expressed as a type restriction. Thus, the context specifications used in PHLIQA1 offer less discrimination power than those used in our system.

There is one other difference regarding λ-context specifications that should be noted here. The context specification *(λb (DM_building b))* will be satisfied by the expression *(λw (DM_warehouse w))*. However, in PHLIQA1 the type DM_building → truthvalue will not match the type DM_warehouse → truthvalue. From this, we see that PHLIQA1 does not use subsumption information in matching λ-context specifications, while our system does.

3.5 Translation and Mapping Selection.

When translating an input sentence containing a comparative expression from EL to DML, the system first applies Trans and Btrans rules to translate the predications that do not represent mappings or base orders. Next, comparison attributes must be selected. The system recognizes comparison attribute ambiguity when there is more than one applicable MTrans rule for a particular EL mapping predicate. We define a **candidate mapping** as any DML mapping that, on the basis of an applicable MTrans rule, can serve as the translation of a mapping in an EL formula. Assume that for an EL predication *(big′ w a)* in a given context there are three applicable MTrans rules translating *big′* to the three DML mappings *DM_w-storage-volume*, *DM_w-storage-area*, and *DM_b-total-area*, respectively. All three of these DML mappings would then be candidates with either VOLUME or AREA as the corresponding base order.

The system examines the semantic lexicon to determine a list of candidate mappings for each EL mapping. A candidate is removed from one of these lists if there is no compatible mapping in the other list. **Compatible mappings** are those that allow the Codomain Agreement Principle to be satisfied, and they are easily identified by examining the base order component of the MTrans rules being used. All of the remaining candidates

[8] This is an extension to the rules used in ATOZ (Ryan, Root, and Olawsky 1988) which do not allow functions as arguments and therefore never need this kind of context checking.

[9] One could perhaps modify the PHLIQA1 world model to contain such subclasses of buildings, but this would eventually lead to a very complex model. It would also be difficult or impossible to keep such a model hierarchical in structure.

in one of the lists are presented to the user who may select a candidate mapping. Next, the semantic processor presents to the user those candidates for the other EL mapping that are compatible with her first choice. She must select one of these remaining candidates as the translation for the second mapping. Based on her choices, two MTrans rules (one for each EL mapping) are applied, and in this way the EL mapping predications are translated to DML formulas. Once this is completed, the processor can easily translate the EL base order to the DML base order listed in both of the MTrans rules it used (with any necessary adjustments in the direction of comparison).

4 COMMENTS AND CONCLUSIONS.

We are currently examining some additional issues. First, once candidate mappings are obtained, how should they be explained to the user? In the present design text is stored along with the declaration of each mapping, and that text is used to describe the mapping to the user. This approach is somewhat limited, especially for adnominal comparatives given their flexibility and the relatively small information content of the **CARDINALITY** mapping. A more general technique would use natural language generation to explain the semantic import of each mapping as applied to its arguments. Perhaps there are compromise approaches between these two extremes (e.g. some kind a pseudo-English explanations).

Second, it seems desirable that the system could work automatically without asking the user which mappings to use. Perhaps the system could choose a mapping, do the query, present the results and then tell the user what interpretation was assumed (and offer to try another interpretation). This works well as long as either (a) the system almost always selects the mapping intended by the user, or (b) the cost of an incorrect choice (i.e. the wasted query time) is small. If the system frequently makes a poor choice and wastes a lot of time, this approach could be quite annoying to a user. Crucial to the success of this automatic approach is the ability to reliably predict the resources required to perform a query so that the risk of guessing can be weighed against the benefits. A similar issue was pointed out by an anonymous reviewer. We noted in Section 1 that for sentence (3) (repeated here as (8))

(8) Are more oranges than apples exported by Mexico?

the comparison could be based on quantity, weight, volume, or value. If the answer is the same regardless of the basis for comparison, a "friendly" system would realize this and not require the user to choose comparison attributes. Unfortunately, this realization is based on extensional rather than intentional equivalence, and hence, the system must perform all four (in this case) queries and compare the answers. The extra cost could be prohibitive. Again, the system must predict query performance resource requirements to know whether this approach is worthwhile for a particular query. See (Olawsky 1989) for more information on further work.

To summarize, we have examined a number of issues associated with the semantic interpretation of comparative expressions and have developed techniques for representing the semantics of CEs and for interacting with the user to resolve comparison attribute ambiguities. These techniques will work for adjectival, adverbial, and adnominal comparatives and for both numerically and non-numerically based comparisons (see (Olawsky 1989) for more on this). We are presently completing the implementation of our approach in Common Lisp using the SunView[10] window system as a medium for user interaction. Most previous techniques for handling lexical ambiguity work best with homonymy since they depend on large semantic differences between the possible interpretations of a lexical item. Our approach, on the other hand, does not depend solely on these semantic differences and handles polysemy well.

5 ACKNOWLEDGEMENTS.

I wish to thank the University of Minnesota Graduate School for supporting this research through the Doctoral Dissertation Fellowship program. I also want to thank Maria Gini, Michael Kac, Karen Ryan, Ron Zacharski, and John Carlis for discussions and suggestions regarding this work.

References

Ballard, Bruce W. June 1988 A General Computational Treatment of Comparatives for Natural Language Question Answering. In: *26th Annual*

[10] SunView is a trademark of Sun Microsystems, Inc.

Meeting of the Association for Computational Linguistics. Buffalo, NY.

Barwise, Jon and Cooper, Robin. 1981 Generalized Quantifiers and Natural Language. *Linguistics and Philosophy* 4(2): 159–219.

Bates, Madeleine and Bobrow, Robert J. 1983 Information Retrieval Using a Transportable Natural Language Interface. In: *Research and Development in Information Retrieval: Proceedings of the Sixth Annual International ACM SIGIR Conference, Bethesda, Md.* New York: 81–86.

Bresnan, Joan W. 1973 Syntax of the Comparative Clause Construction in English. *Linguistic Inquiry* 4(3): 275–343.

Bronnenberg, W. J. H. J.; Bunt, H. C.; Landsbergen, S. P. J.; Scha, R. J. H.; Schoenmakers, W. J.; and van Utteren, E. P. C. 1980 The Question-Answering System PHLIQA1. In: Bolc, L., Ed., *Natural Language Question Answering Systems.* Macmillan.

Cresswell, M. J. 1976 The Semantics of Degree. In: Partee, Barbara, Ed., *Montague Grammar.* Academic Press: 261–292.

Grosz, Barbara; Haas, Norman; Hendrix, Gary; Hobbs, Jerry; Martin, Paul; Moore, Robert; Robinson, Jane; and Rosenschein, Stanley. November 1982 DIALOGIC: A Core Natural-Language Processing System. Tech. Note 270, Artificial Intelligence Center, SRI International, Menlo Park, California.

Hale, Austin. 1970 Conditions on English comparative clause pairings. In: Jacobs, R. A. and Rosenbaum, P., Eds., *Readings in English Transformational Grammar.* Ginn & Co., Waltham, Mass.: 30–50.

Hendrix, Gary G.; Sacerdoti, Earl D.; Sagalowicz, Daniel; and Slocum, Jonathan. 1978 Developing a Natural Language Interface to Complex Data. *ACM Transactions on Database Systems* 3(2): 105–147.

Hirst, Graeme John. May 1984 Semantic Interpretation Against Ambiguity. PhD thesis, Computer Science Dept., Brown University.

Hoeksema, Jack. 1983 Negative Polarity and the Comparative. *Natural Language and Linguistic Theory* 1: 403–434.

Klein, Ewan. 1980 A Semantics for Positive and Comparative Adjectives. *Linguistics and Philosophy* 4: 1–46.

Klein, Ewan. 1982 The Interpretation of Adjectival Comparatives. *Linguistics* 18: 113–136.

Martin, Paul; Appelt, Douglas; and Pereira, Fernando. 1983 Transportability and Generality in a Natural-Language Interface System. In: *Proceedings of the Eighth International Joint Conference on Artificial Intelligence, Karlsruhe, West Germany.* William Kaufmann, Inc., Los Altos: 573–581.

Moore, Robert C. 1981 Problems in Logical Form. In: *Proceedings of the 19th Annual Meeting.* Association for Computational Linguistics, Stanford, California: 117–124.

Olawsky, Duane E. April 1989 The Lexical Semantics of Comparative Expressions in a Muli-Level Semantic Processor. Technical Report CSci TR 89-19, Computer Science Dept., University of Minnesota, Minneapolis, MN.

Pereira, Fernando. 1983 Logic for Natural Language Analysis. Technical Note 275, Artificial Intelligence Center, Computer Science and Technology Division, SRI International, Menlo Park, California. Ph. D. dissertation, Department of Artificial Intelligence, University of Edinburgh.

Pinkham, Jessie Elizabeth. 1985 *The Formation of Comparative Clauses in French and English.* Garland Publishing Inc, New York. Also available from Indiana's University Linguistics Club, Bloomington, IN, August 1982.

Rayner, Manny and Banks, Amelie. June 1988 Parsing and Interpreting Comparatives. In: *26th Annual Meeting of the Association for Computational Linguistics.* Buffalo, NY.

Rusiecki, Jan. 1985 *Adjectives and Comparison in English: A Semantic Study.* Longman Inc., New York.

Ryan, Karen L.; Root, Rebecca; and Olawsky, Duane. February 1988 Application-Specific Issues in NLI Development for a Diagnostic Expert System. In: *Association for Computational Linguistics Second Conference on Applied Natural Language Processing.* Austin, Texas.

Ryan, Karen L. 1983 A Grammar of the English Comparative. PhD thesis, University of Minnesota. Reproduced by Indiana University Linguistics Club, Bloomington Indiana, 1986.

Wilks, Yorick. 1975 An Intelligent Analyzer and Understander of English. *CACM* 18(5): 264–274.