

Exploiting Lexical Regularities in Designing Natural Language Systems

Boris Katz

Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

Beth Levin

Department of Linguistics
Northwestern University
Evanston, IL 60208

*They've a temper, some of them—particularly
verbs: they're the proudest—adjectives you
can do anything with, but not verbs—
however, I can manage the whole lot of them!*

— Lewis Carroll, *Through the Looking-Glass*

Abstract

This paper presents the lexical component of the START Question Answering system developed at the MIT Artificial Intelligence Laboratory. START is able to interpret correctly a wide range of semantic relationships associated with alternate expressions of the arguments of verbs. The design of the system takes advantage of the results of recent linguistic research into the structure of the lexicon, allowing START to attain a broader range of coverage than many existing systems while maintaining modular organization.

1. Introduction

If asked "Did Sally eat?" after having been told that Sally ate a pear, speakers of English would not hesitate to answer "Yes". But we would not expect English speakers to answer "Yes" if asked "Did David dress?" after being told that David dressed the baby. Here the appropriate answer would be "I don't know".

Computational linguists engaged in building Question-Answering systems should find these examples thought-provoking. Two sequences consisting of a statement followed by a question which appear to be parallel syntactically (transitive use of a verb in the statement, intransitive use of the same verb in the question) elicit quite different responses. The simple syntax of these pairs is unlikely to pose a challenge for the parsers used in most existing systems. The problem is that the intransitive uses of the two verbs, *eat* and *dress*, receive very different interpretations. Thus the intransitive use of *eat* found in the question "Did Sally eat?" implies the existence of an understood but unexpressed object that is interpreted as a prototypical type of food or a meal:

(1) Sally ate a pear. \implies Sally ate. (i.e., Sally ate some food.)

The question "Did David dress?" on the other hand does not mean 'Did David dress something one typically dresses?', it means 'Did David dress himself?':

(2) David dressed the baby. $\not\Rightarrow$ David dressed (i.e., David dressed himself.)

Natural language systems should be able to recognize that the relationship between transitive and intransitive *dress* is not the same as that between transitive and intransitive *eat*.

A large number of English verbs have both transitive and intransitive uses. Interchanges parallel to the one described for *eat* are possible with a wide range of verbs:

(3) Jessica typed a letter. Did Jessica type? Yes.

(4) Gabriella swept the floor. Did she sweep? Yes.

(5) Miriam read the book. Did Miriam read? Yes.

But the behavior of the verb *dress* is not exceptional. Another set of verbs including *bathe*, *change*, *shave*, *shower*, and *wash* behave like it. For example, these verbs show the same entailments as *dress*:

(6) Carla bathed the dog. $\not\Rightarrow$ Carla bathed (i.e., Carla bathed herself.)

(7) Jill washed the sweater. $\not\Rightarrow$ Jill washed (i.e., Jill washed herself.)

(8) Peter shaved Tom. $\not\Rightarrow$ Peter shaved (i.e., Peter shaved himself.)

The different relationships between transitive and intransitive uses of verbs cannot be ignored in the design of a natural language system and its lexical component. The most obvious way to handle these relationships is to add information to the lexical entries of each verb with transitive and intransitive uses. While such an approach is viable when a system has a small lexicon, it becomes less tractable as the lexicon grows larger since it requires a tremendous increase in the amount of idiosyncratic information which must be registered in the entry of each verb.

The examples discussed so far illustrate just a few of a wide range of relationships between alternate expressions of the arguments of verbs that must be correctly interpreted by any natural language system that aims at substantial coverage of English. We believe that what is required in order to implement a system that meets these demands is an understanding of English lexical organization. For this reason we draw on recent theoretical linguistic investigations into the lexical knowledge possessed by native speakers of English carried out by the MIT Lexicon Project (Rappaport, Levin, and Laughren [1988], Levin [1985], Hale and Keyser [1986], Levin and Rappaport [to appear]). These studies have established a range of semantic-syntactic interdependencies exhibited by semantically coherent classes of verbs and have identified a number of essential classes of verbs, as well as the central properties characterizing verbs of each type.

The results of this work have been used in the design of a lexical component for the START natural language system developed at the MIT Artificial Intelligence Laboratory (Katz [1988]). In this paper we show how these results allow START to attain a broader range of coverage than most existing systems while maintaining modular organization.

2. Assumptions about Lexical Organization

A verb denotes an action, state, or process involving one or more participants, which we refer to as the *arguments* of the verb. Some verbs may express their arguments in more than one way, sometimes with slightly different semantic interpretations. We say that such verbs participate in *argument alternations*. We have seen that certain verbs have both transitive and intransitive uses, and that the relationship between the uses is not uniform across all verbs; rather, it is a property of the verb involved. We say that such verbs participate in *transitivity alternations*, a subclass of argument alternations. So far, we have seen the *indefinite object* alternation, in (9), and the *reflexive* alternation, in (10):

(9) Sally ate a pear. \Rightarrow Sally ate. (i.e., Sally ate some food.)

(10) David dressed the baby. \nRightarrow David dressed. (i.e., David dressed himself.)

These are only two of about a dozen transitivity alternations found in English. (See Atkins, Kegl, and Levin [1986], Hale and Keyser [1986], Levin [1985] for a description of various other alternations.) A question that a system designer might ask is whether the argument alternations a verb participates in are predictable or whether they are merely an idiosyncratic property of that verb. This question is clearly also of linguistic interest.

Argument alternations, including transitivity alternations, have received considerable attention from theoretical linguists. It turns out that each alternation is associated with particular semantic types of verbs. Thus the verbs participating in the indefinite object alternation are all *activity* verbs; most of them describe typical occupations:¹

(11) drink, eat, file, iron, plow, read, sweep, type, write, ...

The verbs participating in the reflexive alternation are all verbs of *grooming*:

(12) bathe, brush, change, comb, dress, shave, shower, wash, ...

Studies of phenomena such as those described here reveal that English verbs are organized into classes on the basis of shared components of meaning. The members of these classes share certain syntactic properties, specifically properties concerning the possible expressions of their arguments. To state this differently, certain linguistically relevant aspects of a verb's meaning serve as a pointer to its place in the organizational scheme of English verbs. Once this place is identified, various syntactic properties of a verb can be determined.

For linguists working on the lexicon the study of alternations can provide insight into linguistically relevant aspects of meaning due to the interdependency between the syntactic and semantic properties of a verb. (For instance, see Hale and Keyser [1987] for a discussion of another transitivity alternation, the *middle alternation*, from this perspective). For computational linguists this characteristic of lexical organization suggests a modular system design: many lexical properties can be associated with verb classes, and need not be registered in the lexical entries of individual verbs, which can simply indicate class membership. These ideas are incorporated into the design of the START system, as we describe in detail in the remainder of the paper.

¹In certain circumstances, a much wider range of English transitive verbs show an intransitive use with an indefinite object interpretation: when they are used by people whose activity involves the action denoted by the verb to describe the activity. For example, the verb *stuff* does not usually occur without an object, **She stuffs*, but if this verb is used to describe the activity of someone who stuffs pillows in a pillow-factory, then it would be all right to use the verb intransitively.

3. An Overview of the START system

The START natural language system (SynTactic Analysis using Reversible Transformations) consists of two modules which share the same Grammar (see Katz [1980], Katz and Winston [1982]). The *understanding* module analyzes English text and produces a *knowledge base* which incorporates the information found in the text. Given an appropriate segment of the knowledge base, the *generating* module produces English sentences. A user can retrieve the information stored in the knowledge base by querying it in English. The system will then produce an English response.

START has been used by researchers at MIT, Stanford University, and the Jet Propulsion Laboratory for constructing and querying knowledge bases using English. (See, for example, Winston [1982], [1984], Winston, Binford, Katz, and Lowry [1983], Doyle [1984], Katz and Brooks [1987]).

START rearranges the elements of the parse tree it constructs into embedded *ternary expressions* (*T-expressions*) by tying together the three most salient parameters of a sentence, the subject, the object, and the relation between them, <**subject relation object**>. For instance, the sentence

(13) Gabriella might buy some stickers

will result in the T-expression <**Gabriella buy stickers**>.

Certain other parameters (adjectives, possessive nouns, prepositional phrases, etc.) are used to create additional T-expressions in which prepositions and several special words serve as relations.

The remaining parameters—adverbs and their position, tense, auxiliaries, voice, negation, etc.—are recorded in a representational structure called *history*. The history has a *page* pertaining to each sentence which yields the given T-expression. When we index the T-expression in the knowledge base, we cross-reference its three components and attach the history *H* to it. One can thus think of the resulting entry in the knowledge base as a “digested summary” of the syntactic structure of English sentences.

In order to handle embedded sentences, START allows any T-expression to take another T-expression as its subject or object. START can analyze and generate sentences with arbitrarily complex embedded structures.

We conclude our description of START with a brief overview of how the system answers questions. Questions are requests for information from START's knowledge base. In order to answer a question START must translate the question into a T-expression template which can be used to search the knowledge base for T-expressions which contain information relevant to providing an answer to the question. We illustrate the actual question-answering process with an example. Suppose that as a result of analyzing and indexing a text containing sentence (14), the knowledge base contains T-expression (15):

(14) Mary presented Paul with a gift.

(15) <<Mary present Paul> with gift>

Now suppose that a user asks START the following *wh*-question:

(16) Whom did Mary present with a gift?

In the context of (14) the answer is *Paul*. In order to determine this, the system must first turn the question into a T-expression template that can be used to search the knowledge base. The first step in this process is to undo the effects of the *wh*-movement transformation that is used to create English *wh*-questions. To do this, START must find the place in sentence (16) that the *wh*-word *whom* came from and then insert the *wh*-word in this position:

(17) Mary presented *whom* with a gift.

Next the language understanding system leads sentence (17) through the same flow of control as any other declarative sentence and produces the following T-expression which serves as a pattern used to query the knowledge base:

(18) <<Mary present *whom*> with gift>

Treating *whom* as a matching variable, the system feeds T-expression (18) through a matcher in order to determine whether there is anything in the knowledge base that matches (18). The matcher finds the T-expression created from (14):

(19) <<Mary present Paul> with gift>

and the language generation system then uses this T-expression to produce the English response to question (16):

(20) Mary presented Paul with a gift.

START handles *yes-no* questions in a similar fashion. Suppose that START had been asked the *yes-no* question

(21) Did Mary present Paul with a gift?

As in the *wh*-case, START would turn this question into a T-expression template that could be matched against the T-expressions in the knowledge base. The difference between *yes-no* and *wh*-questions is that the T-expression template generated by a *yes-no* question would contain no variables. The question above would generate the template <<Mary present Paul> with gift> which would match against (19), allowing the system to answer:

(22) Yes, Mary presented Paul with a gift.

4. Introducing S-rules

Since T-expressions in the START system are built using the pattern <subject relation object> at every level of embedding, they mimic the hierarchical organization of English sentences. As a consequence, sentences differing in their surface syntax but close in meaning are not considered similar by the system. For example, given sentence (23) as input, START will create an embedded T-expression (24), whereas a near paraphrase, sentence (25), will generate T-expression (26):

(23) Mary presented Paul with a gift

(24) <<Mary present Paul> with gift>

(25) Mary presented a gift to Paul

(26) <<Mary present gift> to Paul>

Speakers of English know that sentences (23) and (25) both describe a transfer of possession. In both sentences, *the gift* is the transferred object, *Paul* is the recipient of this object, and *Mary* is the agent of the transfer, despite different syntactic realizations of some of these arguments. It seems natural that this kind of knowledge be available to a natural language system. However, START, as described so far, does not consider T-expressions (24) and (26), which are associated with these sentences, to be similar.

The difference in the T-expressions becomes particularly problematic when START is asked a question. An example will clarify this point. Suppose the input text contains only one *present* sentence, (27), and the knowledge base contains only the corresponding T-expression, (28):

(27) Mary presented Paul with a gift

(28) <<Mary present Paul> with gift>

Now suppose the user asked the following question:

(29) To whom did Mary present a gift?

Although a speaker of English could easily answer this question after being told sentence (27), START would not be able to answer it. This question presents a problem for START because T-expression (30) produced by question (29) will not match T-expression (28).

(30) <<Mary present gift> to whom>

START is unable to answer such questions because it is unaware of the interactions between the syntactic and semantic properties of verbs. This limitation is a serious drawback since interactions similar to the one just described pervade the English language and, therefore, cannot be ignored in the construction of a natural language system.

The *present* example illustrates that START needs information that allows it to deduce the relationship between alternate realizations of the arguments of verbs. In this instance, we want START to know that whenever A presents B with C, then A presents C to B. We do this by introducing rules that make explicit the relationship between alternate realizations of the arguments of verbs. We call such rules *S-rules* (where *S* stands for both *Syntax* and *Semantics*). Here is the S-rule that solves the problem caused by the verb *present*:

(31) *Present* S-rule

If <<subject present object1> with object2>

Then <<subject present object2> to object1>

S-rules are implemented as a rule-based system. Each S-rule is made up of two parts, an antecedent (the IF-clause) and a consequent (the THEN-clause). Each clause consists of a set of templates for T-expressions, where the template elements are filled by variables or constants. For example, the *Present* S-rule contains three variables, *subject*, *object1*, *object2*, which are used to represent the noun phrases in the T-expressions. This rule also contains three constants, **present**, **with**, and **to**, shown in boldface. The *Present* S-rule will apply only to T-expressions which involve the verb *present* and which meet the additional structural constraints.

S-rules operate in two modes: *forward* and *backward*. We describe both modes of operation briefly, although in this paper we concentrate on S-rules operating in the backward mode, since this mode is most useful in a Question-Answering natural language system.

When triggered by certain conditions, S-rules in the forward mode allow the system to intercept T-expressions produced by the understanding module, transform or augment them in a way specified by the rule, and then incorporate the result into the knowledge base. For instance, if the *Present* S-rule is used in the forward mode, as soon as its antecedent matches T-expression (32) produced by the understanding module, it creates a new T-expression (33) and then adds it to the knowledge base:

(32) <<Mary present Paul> with gift>

(33) <<Mary present gift> to Paul>

Now question (29) can be answered since T-expression (30) associated with this question matches against T-expression (33). The generating module of START responds:

(34) Mary presented a gift to Paul.

All additional facts produced by the forward S-rules are instantly entered in the knowledge base. The forward mode is especially useful when the information processed by START is put into action by another computer system because in such a situation START ought to provide the interfacing system with as much data as possible.

In contrast, the backward mode is employed when the user queries the knowledge base. Often for reasons of computational efficiency, it is advantageous not to incorporate all inferred knowledge into the knowledge base immediately. S-rules in the backward mode trigger only when a request comes in which cannot be answered directly, initiating a search in the knowledge base to determine if the answer can be deduced from the available information. For example, the *Present* S-rule used in the backward mode does *not* trigger when sentence (27) is read and T-expression (28) is produced by START. The S-rule triggers only when question (29) is asked since this question cannot be answered directly.

5. The Lexical Component of START

In order to understand an English sentence, the START system needs to have morphological, syntactic, and semantic information about the words in the sentence. All the words that the system is aware of, along with information about their part of speech, inflection, gender, number, *etc.* are stored in the *Lexicon*. Virtually every branch of START uses the Lexicon to accomplish its task. In this section we discuss the way in which the Lexicon extends the system's ability to deal with semantic-syntactic interdependencies. We show that the Lexicon provides a place where a verb's membership in a semantic class can be registered, allowing more general S-rules to be stated.

To examine how lexical information about verb classes may be utilized by the S-rules, we introduce another argument alternation in English, the *property-factoring* alternation (Van Oosten [1980]). Consider the following sentence:

(35) Paul surprised the audience with his answer.

An English speaker knows that sentence (35) can be paraphrased as:

(36) Paul's answer surprised the audience.

Notice that in (35), the subject brings about the emotional reaction (*surprise*) by means of some property expressed in the *with* phrase. Sentence (36) describes the same emotional reaction as in (35) but in (36) the property and its possessor are collapsed into a single noun phrase.

Suppose that after sentence (35) is typed into the computer, we ask:

(37) Did Paul's answer surprise the audience?

While a speaker of English would know that the answer to this question is *Yes*, this reply is not obvious to START since T-expressions related to sentence (35) and question (37) are very different.²

(38) <<Paul surprise audience> with answer>

(39) <answer surprise audience>

Extending the approach taken to the example with the verb *present* in section 4, we could formulate a simple S-rule that could be used to answer question (37). The *Surprise* S-rule (40), like the *Present* S-rule, makes explicit the relationship between the alternate realizations of the arguments of the verb *surprise*:

(40) *Surprise* S-rule

If <<subject surprise object1> with object2>
Then <object2 surprise object1>

In taking this approach we are explicitly associating the property-factoring alternation with the verb *surprise*; we are assuming that it is an idiosyncratic property of the verb.

Formulating a special purpose S-rule which applies only to the verb *surprise* does not seem to be the best solution to the problem. *Surprise* is only one of many verbs which exhibit the property-factoring alternation. This alternation holds of a large class consisting of over one hundred verbs, among them

(41) anger, annoy, embarrass, frighten, impress, please, ...

For example:

(42) Miriam amused Jessica with her performance.

(43) Miriam's performance amused Jessica.

(44) Gabriella scared the turtle with a sudden movement.

(45) Gabriella's sudden movement scared the turtle.

These verbs also share a certain semantic property: they all denote *emotional reactions*. For this reason we identify a class of *emotional-reaction* verbs and say that the property of the verb *surprise* responsible for the alternation shown in (35) and (36) holds for all verbs that comprise the *emotional-reaction* class.³

Once we have tied the ability to participate in the property-factoring alternation to a particular class of verbs, we no longer need to indicate this property in the lexical entry of each verb in the class or write verb-specific S-rules, such as the *Surprise* S-rule. Rather, we can associate the alternation with the class and then simply indicate in the lexical entry of a verb whether it belongs to this class. That is, we augment a verb's lexical entry with an indication of its semantic class membership. For instance, we would register in the entry for *surprise* that it is a member of the *emotional-reaction* class.⁴

(46) (surprise :verb :emotional-reaction)

Now instead of writing a number of verb-specific S-rules, we can write a single general S-rule which triggers not only on the verb *surprise*, but on any verb from the *emotional-reaction* class:

(47) *Property-factoring* S-rule

If <<subject verb object1> with object2>
Then <object2 verb object1>
Provided verb \in *emotional-reaction* class

The revised S-rule contains a PROVIDED clause which specifies the class of verbs to which the rule applies, ensuring that it applies to the *emotional-reaction* verbs.

When question (37) is asked, the *Property-factoring* S-rule (used in the backward mode) will trigger, since the T-expression

(48) <answer surprise audience>

²To simplify the exposition we do not show the T-expression describing the relation between the property (*answer*) and its possessor (*Paul*).

³These verbs have been the subject of extensive study in the linguistic literature because of these and other characteristic properties that set this class apart. (See Postal [1971], Pesetsky [1987], Belletti and Rizzi [1986], Grimshaw [to appear], and many others). This class of verbs which take the experiencer argument (the person experiencing the emotions) as object should be distinguished from a second class of verbs of psychological state which take the experiencer argument as subject. The latter class which includes verbs like *admire*, *detest*, *esteem*, *hate*, and *like* will not be discussed in this paper.

⁴Irrelevant details have been suppressed in this lexical entry.

produced by the question matches the THEN-part of the rule, and furthermore, the verb *surprise* belongs to the emotional-reaction class. The correct answer to question (37) is deduced when the appropriately instantiated IF-part of the S-rule is matched to T-expression (38) found in the knowledge base. Here is how START responds:

(49) Yes, Paul's answer surprised the audience.

The PROVIDED restriction of S-rule (47) not only allows the rule to apply to verbs of the appropriate semantic type, but it also prevents the rule from applying to verbs that do not display the property-factoring alternation. For instance, the verbs *surprise* and *present* can express their arguments in a similar fashion—both are found in the context [NP V NP with NP], but they differ in the other realizations of their arguments. Specifically *present* does not participate in the property-factoring alternation, as (50) shows, nor does *surprise* participate in the alternation that *present* participates in, as (51) shows:

(50) Mary presented Paul with a gift.

*Mary's gift presented Paul.

(51) Paul surprised the audience with his answer.

*Paul surprised his answer to the audience.

In the absence of the PROVIDED clause, the Property-factoring S-rule could potentially misapply to verbs like *present*.

The *surprise* example shows how the addition of information about semantic class membership to verb entries allows the system to handle a particular phenomenon (or lexical property) common to all verbs in a particular class, with the help of a single S-rule. We refer to this approach as the *verb class* approach.

We could have achieved the same effect in another way: instead of specifying verb class membership in the entry of each verb, we could have explicitly registered the lexical properties that apply to the verb (or the names of the corresponding S-rules). Taking this approach, the lexical entry for *surprise* would indicate *property-factoring* instead of *emotional-reaction* but would otherwise be unchanged.

(52) (**surprise** :verb :property-factoring)

This approach could allow us to dispense with the PROVIDED clause in the S-rules since the lexical entry of a verb would be tagged with the set of S-rules that could apply to that verb.

However, the verb class approach has a clear advantage over the alternative *lexical property* approach when more than one property is involved. Typically each semantic class of verbs has a number of properties associated with it, which must ultimately be handled by a natural language system.

If we take the lexical property approach, whenever we add a new lexical property, we would need to write an S-rule for this new property, and we would then have to add the property to the lexical entry of each member of the class of verbs that exhibit this property.

In contrast, if we take the verb class approach, it is easy to extend the system to handle new properties of a class of verbs. All that is required is the addition of the appropriate S-rule, formulated so that it triggers on this class of verbs. There is no need to alter the lexical entries of the members of the class in any way if the lexical entry of each verb in the class indicates that it is a member of this class. Thus the verb class approach, unlike the lexical property approach, allows a more modular system design; this in turn allows the coverage of the system to be extended more easily.

To illustrate that each class of verbs is characterized by a whole cluster of properties, we survey just a few of the properties associated with the emotional-reaction verbs. They participate in several transitivity alternations—the *middle* alternation in (53) and (54) (see Keyser and Roeper [1984], Hale and Keyser [1987], among others) and the *null-object* alternation in (55) and (56) (Rizzi [1986]):

(53) Dogs frighten little children.

(54) Little children frighten easily.

(55) Thunder never fails to frighten people.

(56) Thunder never fails to frighten.

The subject of the middle use (54) bears the same semantic relation to the verb as the object of the transitive use (53), while the subject of the null-object use (56) bears the same semantic relation to the verb as the subject of the transitive use (55). The null-object alternation, like the indefinite object alternation, involves an unexpressed but understood object in the intransitive variant. However, the understood object here is interpreted as "people".

Emotional-reaction verbs are also set apart by another property: they only have passive nominals. Thus, *the children's disappointment* (passive interpretation), not that they caused disappointment (active interpretation). Compare the interpretation of this nominal to that of the nominal related to a verb like *describe*: *Gene's description* can refer to the description that Gene gave (active interpretation) or to a description that someone gave of Gene (passive interpretation).

We return to the example involving *present* discussed in the previous section. The alternation manifested by the verb *present* is exhibited by the members of a small class of verbs which also includes *entrust*, *furnish*, *supply*, and *trust*. The *Present* S-rule in (31) can be generalized so that it will apply to the entire class of verbs like *present*, which we name the *present* class.

(57) *Presentation* S-rule

If <<subject verb object1> with object2>
 Then <<subject verb object2> to object1>
 Provided verb ∈ *present* class

Now suppose that after typing (58) we ask question (59):

(58) Mark supplied the restaurant with fresh vegetables.

(59) Did Mark supply fresh vegetables to the restaurant?

Since the verb *supply* is a member of the *Present* class, the S-rule in (57) applies and START will be able to answer the question correctly:

(60) Yes, Mark supplied fresh vegetables to the restaurant.

The examples described in this section show how the transparent syntax of the S-rules coupled with the information about verb class membership provided by the Lexicon facilitates fluent and flexible dialog between the user and the language processing system.

6. Employing S-Rules: Additional Examples

In this section we present additional dialogues that START can handle. These interchanges are chosen to illustrate the use of S-rules as well as the range of coverage of the START system. In the process we introduce S-rules that handle several more semantic-syntactic interdependencies, including the indefinite object and reflexive alternations discussed in the introduction to the paper.

At the beginning of the paper we presented two short interchanges between two speakers of English and asked what it would take for a natural language system to be able to participate in these interchanges. We repeat the examples below:

(61) *Speaker 1*: Sally ate a pear.

Speaker 1: Did Sally eat?

Speaker 2: Yes.

(62) *Speaker 1*: David dressed the baby.

Speaker 1: Did David dress?

Speaker 2: I don't know.

These examples were used to illustrate some of the semantic-syntactic interdependencies characteristic of verbs that participate in transitivity alternations. Verbs that have both transitive and intransitive uses differ in the interpretations associated with their intransitive uses, as these interchanges were intended to show. S-rules allow START to answer questions that draw on knowledge of the semantic-syntactic interdependencies that are manifested in transitivity alternations.

The two transitivity alternations relevant to the interchanges above are the indefinite object alternation (with *eat*) and the reflexive alternation (with *dress*). When used intransitively, verbs that participate in the indefinite object alternation describe actions where there is an implied object which is understood to be something that is a typical object of the verb. Thus:

(63) Sally ate a pear. \implies Sally ate. (i.e., Sally ate some food or a meal.)

We can formulate an S-rule that makes explicit the relationship between the transitive and the intransitive use of such verbs.

(64) *Indefinite-object* S-rule

If <subject verb object>
Then <subject verb>
Provided verb \in *activity/occupation* class

Now if START were asked "Did Sally eat?" after having been told that Sally ate a pear, START would be able to answer "Yes", assuming that *eat*'s lexical entry indicates that it is a member of the activity/occupation class.

In order to handle the *dress* interchange, START needs an S-rule that captures the properties of the reflexive alternation shown by verbs of grooming. The intransitive use of a verb that participates in this alternation implies that the subject performed the action denoted by the verb on himself or herself.

(65) David dressed. \implies David dressed himself.

We can capture the special interpretation associated with the intransitive use of a verb like *dress* by means of an S-rule:

(66) *Reflexive* S-rule

If <subject verb>
Then <subject verb subject>
Provided verb \in *grooming* class

Now if asked "Did Jeff dress himself?", after being told that Jeff dressed, the system would answer "Yes" since the lexical entry of *dress* indicates that it is a verb of grooming. START will also be able to handle the following more complex interchange:

(67) *Input*: Ann dressed and Mary dressed the baby.

Question: Who dressed herself?

START: Ann dressed herself.

Question: Who dressed the baby?

START: Mary dressed the baby.

Question: Did Mary dress?

START: I don't know.

In a more complex situation, S-rules are allowed to trigger each other and to ask each other for help. To exemplify this, we

introduce two more S-rules. These rules are used to capture the properties of a class of verbs that has received substantial attention in the linguistics literature (see Anderson [1971], Fraser [1971], Schwarz-Norman [1976], Jeffries and Willis [1984], Rapaport and Levin [1986], among others). We refer to this class as the *spray/load* class after two prototypical members. The verbs in this class describe actions that involve putting substances or materials on surfaces (e.g. *spray*) or in containers (e.g. *load*). What distinguishes one member of the *spray/load* class from another is the manner in which this action is performed. Some members of this class are listed below:

(68) load, pack, pile, smear, spray, spread, stack, stuff, wrap, ...

All the members of this class display an argument alternation, the *locative* alternation; they are found in two syntactic frames, as illustrated below:

(69) Miriam sprayed paint on the wall. (*locative* variant)

(70) Miriam sprayed the wall with paint. (*with* variant)

(71) Jan loaded hay on the truck.

(72) Jan loaded the truck with hay.

Although the sentences in each of these pairs might appear at first glance to be paraphrases, they are not. For instance, in the first pair, only (69), the *locative* variant, may be used to describe a situation where a small portion of the wall is covered with paint as a result of the spraying (the *partitive* interpretation), while sentence (70), the *with* variant, necessarily implies that the wall is entirely covered with paint as a result of the action (the *holistic* interpretation). The sentences in each pair are near paraphrases in the sense that the truth of the *with* variant entails the truth of the *locative* variant, but not vice versa.

Before we can formulate an S-rule that captures the relation between the two variants, we need to look more closely at the *locative* variant. When used in the *locative* variant, *spray/load* verbs are found with a wide range of *locative* prepositions (the set of prepositions indicating spatial relationships).

(73) Miriam sprayed paint *on/under/around* the table.

When the *with* variant of a *spray/load* verb is paraphrased by the *locative* variant, typically only one of the range of *locative* prepositions is possible; the choice is a function of the verb involved. The verbs *spray* and *load* both involve the preposition *on* in the *locative* variant, but the verb *stuff* requires the preposition *into*, while the verb *wrap* requires *around*:

(74) Jessica stuffed the pillow with feathers.

(75) Jessica stuffed the feathers *into* the pillow.

(76) Gabriella wrapped the package with paper.

(77) Gabriella wrapped paper *around* the package.

It appears that each *spray/load* verb is associated with a default *locative* preposition that must be indicated in its lexical entry:

(78) (**spray** :verb :spray-load :default-preposition **on**)

(79) (**wrap** :verb :spray-load :default-preposition **around**)

(80) (**stuff** :verb :spray-load :default-preposition **into**)

We can now write an S-rule that captures the relation between the *with* and *locative* variants of the *locative* alternation:

(81) *Holistic/partitive* S-rule

If <<subject verb object1> **with** object2>
Then <<subject verb object2> prep_{LOC} object1>
Provided verb \in *spray/load* class

In this S-rule the variable 'prep_{LOC}' is instantiated with the default *locative* preposition associated with the verb the rule is applying to.

Using the Holistic/partitive S-rule, START is easily able to handle the following interchanges:

- (82) *Input*: Matilda stuffed the suitcase with books.
Question: Did Matilda stuff the books into the suitcase?
START: Yes, Matilda stuffed the books into the suitcase.
- (83) *Input*: Miriam wrapped the book with paper.
Question: Who wrapped the paper around the book?
START: Miriam wrapped the paper around the book.

To illustrate how S-rules can trigger each other, we introduce an S-rule motivated by another property of *spray/load* verbs. We have seen that the *with* variant of a *spray/load* verb entails the locative variant. In addition, the locative variant, which describes the placement of some substance in a container or on some surface, implies that the substance will be in the container or on the surface. That is, sentence (84) entails sentence (85):

- (84) Jan loaded hay on the truck.
(85) Hay was on the truck.

The following S-rule can be used to capture this property of *spray/load* verbs.⁵

- (86) *Resulting Location* S-rule
- If** <<subject verb object1> prep_{LOC} object2>
Then <<subject be> prep_{LOC} object2>
Provided verb ∈ *spray/load* class

We attribute the entailment relationship between the two variants described by the Holistic/partitive S-rule to a speaker's linguistic knowledge, while the entailment relationship described by the Resulting Location S-rule reflects real world knowledge. We introduced the Resulting Location S-rule in order to show how one S-rule can operate on the output of another. The Holistic/partitive S-rule and the Resulting Location S-rule together allow sentence (88) to be deduced from sentence (87):

- (87) Jan loaded the truck with hay.
(88) Hay was on the truck.

By using both these S-rules, the system can handle the following interchange:

- (89) *Input*: Jan loaded the truck with hay.
Question: Was there hay on the truck?
START: Yes, there was hay on the truck.

The syntactic component of START is able to undo the effects of the There-Insertion rule in the question, translating it into the T-expression

- (90) <<hay be> on truck>

This T-expression does not match anything in the knowledge base, so START tries to apply S-rules. The Resulting Location S-rule used in the backward mode can apply to this T-expression, suggesting that START then searches for the T-expression that matches template (91) below, where the variable *verb* is restricted to members of the *spray/load* class.

- (91) <<subject verb hay> on truck>

This T-expression also does not match against anything in the knowledge base, but it triggers the Holistic/partitive S-rule,

⁵The PROVIDED clause in the Resulting Location S-rule restricts this rule to the *spray/load* verbs. Actually this S-rule applies to a larger set of verbs, the set of causative verbs of change of location, including *put*, *place*, *insert* and transitive *move*, *slide*, *drop*, as well as the *spray/load* verbs. For instance, if Rebecca put the books on the table, then the books are on the table. We have given a restrictive formulation of the PROVIDED clause in this S-rule since we do not want to discuss the implications of introducing a hierarchy of verb class types for the operation of S-rules.

which when applied produces

- (92) <<subject verb truck> with hay>

Finally, T-expression (92) matches with the T-expression

- (93) <<Jan load truck> with hay>

obtained from the *Input* sentence in (89), allowing START to generate the appropriate answer.

Below we show some further examples of interchanges successfully handled by START through the use of additional S-rules. These examples involve verbs of *manner of motion*, listed in (94), and verbs of *creation*, listed in (97):

- (94) climb, cross, fly, gallop, jump, march, swim, walk, ...

- (95) *Input*: Albert jumped over the fence.

Question: Did he jump the fence?

START: Yes, Albert jumped the fence.

- (96) *Input*: Candy climbed up the mountain.

Question: What did Candy climb?

START: Candy climbed the mountain.

- (97) bake, carve, crochet, hammer, knit, sew, weave, ...

- (98) *Input*: Jessica baked the dough into round loaves.

Question: What did Jessica bake round loaves from?

START: Jessica baked round loaves from the dough.

- (99) *Input*: Miriam carved wood into a doll.

Question: Out of what did Miriam carve the doll?

START: Miriam carved the doll out of the wood.

At any given moment many S-rules may be hidden in the computer's memory examining the output flow generated by START and waiting for their turn to participate in the deduction process. S-rules fundamentally expand the power of our system; they open a window into the intricate world of semantic-syntactic interactions.

7. Lexical Acquisition

A natural language system must be built in such a way that it is easy to expand its coverage, not only by increasing the size of the lexicon but also by adding to the set of different phenomena covered by its grammar. Due to the large number of semantic-syntactic interdependencies, increasing the coverage of a system's grammar might seem to be prohibitively expensive. It would require meddling with the entries of every verb in the Lexicon, in order to register its behavior with respect to the new phenomenon. But once a phenomenon is associated with one or more verb classes, it need *only* be associated with these classes. There is no need to tamper with the entries of the individual verbs or construct verb-specific S-rules, as long as the lexical entries of verbs indicate class membership. Thus the problem of incorporating new phenomena is considerably simplified.

The process of *lexical acquisition* (adding new words to the Lexicon and specifying the relevant information about them) is very simple in START. Introducing a new lexical item amounts to little more than appending it to a list of similar words, adding a few idiosyncratic features when necessary. For example, if we wanted to add the verb *annoy* to START's lexicon, we would simply have to add the verb together with an indication that it is a member of the emotional-reaction class.

- (100) (**annoy** :verb :emotional-reaction)

The lexical entry would not need to contain an explicit indication that this verb participates in the property-factoring alternation since the S-rule representing this alternation makes explicit that this property holds of all members of the emotional-

reaction class. The class membership indication in *annoy*'s lexical entry would allow the S-rules that apply to the emotional-reaction class in general to apply to this verb in particular, so that START will be able to handle sequences such as the following:

- (101) *Input*: The dog annoyed the guests with its loud barking.
Question: Whom did the dog's loud barking annoy?
START: The dog's loud barking annoyed the guests.

The acquisition of S-rules is equally simple in the START system due to a special component that allows START to infer S-rules from examples. Adding a new S-rule to the system requires typing in a set of English sentences which capture a specific instance of the rule. For instance, a pair of declarative sentences (such as (35) and (36)), which exemplify the property-factoring alternation, can be used by START to infer the related S-rule. To do this, START analyzes the sentences, queries the user for additional information regarding elements of corresponding T-expressions (ascertaining whether they are matching variables, constants, or predicates), and then builds and generalizes the S-rule automatically.

Careful examination of English verb classes (see Levin [to appear]) combined with the effective employment of S-rules allows the system to reduce to a minimum the amount of idiosyncratic syntactic and semantic information in the Lexicon. All this makes the system transportable; that is, it is easily adaptable to new domains.

8. Conclusion

The addition of a component that explicitly encodes verb classes and their characteristic properties, enables the START system to handle a wide range of phenomena reflecting semantic-syntactic correspondences that are characteristic of English verbs. By factoring properties that belong to whole classes of verbs out of the entries of individual verbs and letting these entries simply designate the verb's class membership, we do more than merely simplify entries. We facilitate the addition of new words to the lexicon and make it easier to extend the system's coverage of linguistic phenomena.

Acknowledgments

We are grateful to Jane Simpson, Misha Katz, and Tom Marill for helpful comments and suggestions concerning this paper, and to Jeff Palraucci who contributed significantly to the Question-Answering part of the system.

This paper describes research done at the Massachusetts Institute of Technology. Support for Katz's work was provided in part by the Advanced Research Projects Agency under Office of Naval Research contract N0014-85-K-0124. Support for Levin's work was provided in part by a grant to the Lexicon Project of the MIT Center for Cognitive Science from the System Development Foundation.

References

1. S.R. Anderson, "On the Role of Deep Structure in Semantic Interpretation," *Foundations of Language* 7, 387-396, 1971.
2. B.T. Atkins, J. Kegl, and B. Levin, "Explicit and Implicit Information in Dictionaries," Lexicon Project Working Papers 12, Center for Cognitive Science, MIT, Cambridge, MA, 1986.
3. A. Belletti and L. Rizzi, "Psych-Verbs and Th-Theory," Lexicon Project Working Papers 13, Center for Cognitive Science, MIT, Cambridge, MA, 1986. To appear in *Natural Language and Linguistic Theory*.
4. R.J. Doyle, "Hypothesizing and Refining Causal Models," M.I.T. Artificial Intelligence Laboratory Memo No. 811, December 1984.
5. B. Fraser, "A Note on *Spray Paint* Cases," *Linguistic Inquiry* 2, 603-607, 1971.
6. J. Grimshaw, *Argument Structure*, MIT Press, Cambridge, MA, to appear.
7. K.J. Hale and S.J. Keyser, "Some Transitivity Alternations in English," Lexicon Project Working Papers 7, Center for Cognitive Science, MIT, Cambridge, MA, 1986.
8. K.J. Hale and S.J. Keyser, "A View from the Middle," Lexicon Project Working Papers 10, Center for Cognitive Science, MIT, Cambridge, MA, 1987.
9. L. Jeffries and P. Willis, "A Return to the *Spray Paint* Issue," *Journal of Pragmatics* 8, 715-729, 1984.
10. B. Katz, "A Three-step Procedure for Language Generation," M.I.T. Artificial Intelligence Laboratory Memo No. 599, December 1980.
11. B. Katz, "Using English for Indexing and Retrieving," *RIAO-88 Conference on User-oriented Content-based Text and Image Handling*, Cambridge, MA, 1988.
12. B. Katz and R.N. Brooks, "Understanding Natural Language for Spacecraft Sequencing," *Journal of the British Interplanetary Society*, vol. 40, no. 10, 1987.
13. B. Katz and P.H. Winston, "A Two-way Natural Language Interface," in *Integrated Interactive Computing Systems*, edited by P. Degano and E. Sandewall, North-Holland, Amsterdam, 1982.
14. S.J. Keyser and T. Roeper, "On the Middle and Ergative Constructions in English," *Linguistic Inquiry* 15, 381-416, 1984.
15. B. Levin, "Introduction," in *Lexical Semantics in Review*, edited by B. Levin, Lexicon Project Working Papers 1, Center for Cognitive Science, MIT, Cambridge, MA, 1985.
16. B. Levin, "Approaches to Lexical Semantic Representation," in *Automating the Lexicon*, edited by D. Walker, A. Zampolli, and N. Calzolari, MIT Press, Cambridge, MA, to appear.
17. B. Levin and M. Rappaport, *Unaccusativity: at the Syntax-Semantics Interface*, MIT Press, Cambridge, MA, to appear.
18. D. Pesetsky, "Binding Problems with Experiencer Verbs," *Linguistic Inquiry* 18, 126-140, 1987.
19. P. Postal, *Cross-Over Phenomena*, Holt, Rinehart, and Winston, N.Y., 1971.
20. M. Rappaport and B. Levin, "What to Do with Theta-Roles," Lexicon Project Working Papers 11, Center for Cognitive Science, MIT, Cambridge, MA, 1986.
21. M. Rappaport, B. Levin, and M. Laughren, "Niveaux de representation lexicale," *Lexique* 7, 1988. An English translation "Levels of Lexical Representation" appears as Lexicon Project Working Papers 20, Center for Cognitive Science, MIT, 1987.
22. L. Rizzi, "Null Objects in Italian and the Theory of *pro*," *Linguistic Inquiry* 17, 501-557, 1986.
23. L. Schwartz-Norman, "The Grammar of 'Content' and 'Container,'" *Journal of Linguistics* 12, 279-287, 1976.
24. J. Van Oosten, "Subjects, Topics and Agents: Evidence from Property-factoring," *Proceedings of the Berkeley Linguistics Society* 6, Berkeley, CA, 1980.
25. P.H. Winston, *Artificial Intelligence*, Addison-Wesley, Reading, MA, 1984.
26. P.H. Winston, "Learning New Principles from Precedents and Exercises," *Artificial Intelligence*, vol. 19, no. 3, 1982.
27. P.H. Winston, T.O. Binford, B. Katz, M.H. Lowry, "Learning Physical Descriptions from Functional Definitions, Examples, and Precedents," *National Conference on Artificial Intelligence*, Washington, D.C., 1983.