A Cross-Linguistic Approach to Translation

Bonnie J. Dorr
MIT Artificial Intelligence Laboratory, Cambridge, MA
E-mail: bonnie@reagan.ai.mit.edu

1 Introduction

This paper describes UNITRAN, an implemented machine translation system that translates Spanish, English, and German bidirectionally.[1] The primary characteristic of UNITRAN is that it operates *cross-linguistically (i.e.,* uniformly across all languages), while still accounting for knowledge that is specific to each language. The task of cross-linguistic translation is difficult because there are several types of phenomena within any given language; moreover, the number of ways in which these phenomena can be exhibited is potentially enormous across different source and target languages.

Consider the following translation from English to Spanish:

(1) Juan forzó la entrada al cuarto (John forced entry to the room) ⇒
John broke into the room

In this example, the source-language sentence diverges both syntactically and lexically from the target-language sentence. The syntactic divergence shows up in the realization of a single verbal object *(room)* in English, even though the Spanish sentence realizes two verbal objects *(entrada* and *cuarto).* The lexical divergence shows up in the realization of the main action as the single verb *break* in English even though the composite form *forzar la entrada* (literally, *force entry)* is used in Spanish. The UNITRAN system solves these types of divergences by providing a principle-based framework within which lexical-semantic information is abstracted to a level that is distinct from that of syntactic information; sentences are then translated on the basis of an interaction between these two levels.

The overall design of UNITRAN is shown in figure 1. The syntactic level of the system is based on a set of linguistic principles, along with their associated parameters, drawn from *government-binding* (GB) theory (see Chomsky (1981, 1982)). This level consists of the information necessary to accept or produce grammatically correct sentences. The lexical-semantic portion of the system is based on theories of *lexical conceptual structure* (LCS) (see Jackendoff (1983, 1990), Hale & Laughren (1983), and Hale & Keyser (1986)). This level consists of the information necessary to provide an underlying conceptual form (the LCS) and to match this structure to the appropriate target-language lexical items. The syntactic level of processing will be discussed briefly in the next section, and the rest of the paper will focus on the lexical-semantic level of processing and the types of problems that are solved within the LCS framework.

What makes the task of principle-based translation difficult is the requirement that the translator process many types of *language-specific* phenomena while still maintaining *language-independent* information about the source and target languages. For example, it is conceivable that the system might translate a Spanish sentence incorrectly on the basis of the knowledge it has for translating English sentences. Consider the Spanish sentence (2):

(2) Qué golpeó Juan (What did John hit)

[1] The name UNITRAN stands for UNIversal TRANslator; i.e., the system serves as the basis for translation across a variety of languages, not just two languages or a family of languages. To date, the system operates only on the three languages mentioned, but plans are currently underway for the addition of Warlpiri, a native aborigine language of Australia.

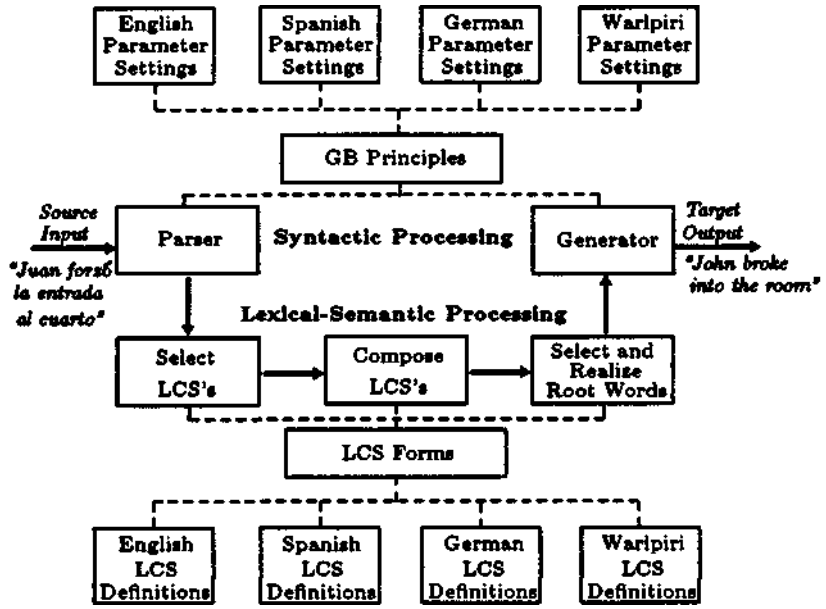

Figure 1: Overall Design of UNITRAN, Dorr (1987, 1989a, 1989b, 1990)

If the translator were to use its syntactic knowledge of English to translate this sentence, it would understand the sentence to mean *what hit John* (*i.e.,* the *agent* and *goal* roles would be reversed). Thus, it is crucial that the translator know certain language-specific information (*e.g.,* the word-order permitted by a particular language) so that it can provide an appropriate structural realization of this sentence; in addition, the translator must know certain language-independent information (*e.g.,* the roles that are introduced by the *hit* action) so that it can assign the appropriate interpretation regardless of how the particular languages structurally realize the sentence.

Given that these two types of knowledge (language-specific and language-independent) are required to fulfill the translation task, one approach to machine translation is to provide a common language-independent representation that acts as a *pivot* between the source and target languages, and to provide a language-specific mapping between this form and the input and output of each language. In the UNITRAN system, the pivot form is the composed LCS that underlies the source- and target-language sentences. This pivot approach to translation is called *interlingual* because it is based on an underlying form derived on the basis of universal *principles* that hold across all languages. Within this framework, the UNITRAN system handles the distinctions among languages by referring to the settings of *parameters* associated with the universal principles. For example, there is a GB principle that is concerned with the absence or presence of the subject in a sentence. The parameter that is associated with this principle, called the *null subject* parameter, is set to *yes* for Spanish (also, Italian, Hebrew, *etc.*) but no for English and German (also French, Warlpiri, *etc*.). This accounts for the possibility of a missing subject in Spanish and the impossibility of a missing subject in English and German.

Setting the null subject parameter in the UNITRAN system is done through a simple menu operation as shown in figure 2. This parameter setting accounts for the word order variation in example (2). Because null subject languages have the property that subjects may freely invert into post-verbal position, the noun phrase *Juan* is taken to be the subject of the Spanish sentence even though it would be taken to be the object in the structurally equivalent English sentence.

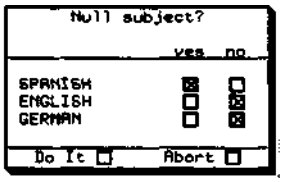

Figure 2: Choosing the Null Subject Setting in UNITRAN

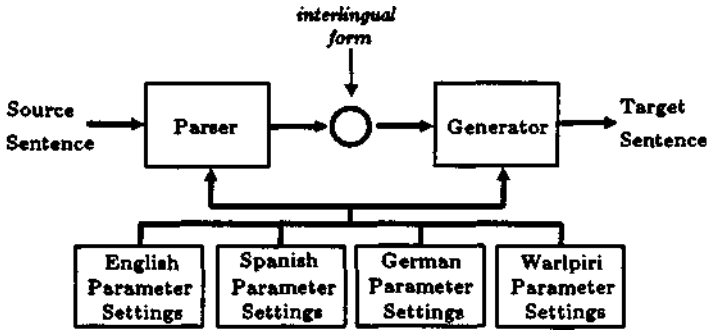

Figure 3: Design of the Syntactic Component in UNITRAN

Both of the levels shown in figure 1 operate on the basis of language-independent and language-specific knowledge. Within the syntactic level, the language-independent and language-specific information are supplied by the GB principles and parameters, respectively. Within the lexical-semantic level, the language-independent and language-specific information are supplied by a set of general lexical-semantic forms and the associated LCS definitions for each language, respectively. (A more detailed presentation of the LCS descriptions will be provided in section 3.) The interface between the syntactic and semantic levels allows the source-language structure to be mapped systematically to the conceptual form, and it allows the target-language structure to be realized systematically from lexical items derived from the conceptual form. This work represents a shift away from complex, language-specific syntactic translation without entirely abandoning syntax. Furthermore, the work moves toward a model that employs a well-defined lexical conceptual representation without relying entirely on semantics.

The next section will provide a brief description of the syntactic component of the UNITRAN system. Section 3 will describe the lexical conceptual component of UNITRAN, and it will present a mapping between the semantic and syntactic levels. While the mapping between these two levels is perhaps the most important advance of the UNITRAN system, another major contribution of the system is its ability to achieve two crucial machine-translation operations, *lexical selection* and *syntactic realization,* despite the potential for syntactic and lexical divergences. The execution of these two operations will be described in section 4. In section 5, we will see that the UNITRAN system differs from other translation systems in that it applies cross-linguistically, and it relies on compositionality and lexical-semantics/syntax abstraction in order to overcome translation divergence problems. Finally, section 6 will demonstrate the translation process for example (1), showing how the system is able to select and realize the appropriate target-language word *break* as the translation of *forzar* (literally *force)* in (1) despite the fact that *break* is not the literal translation of *forzar.*

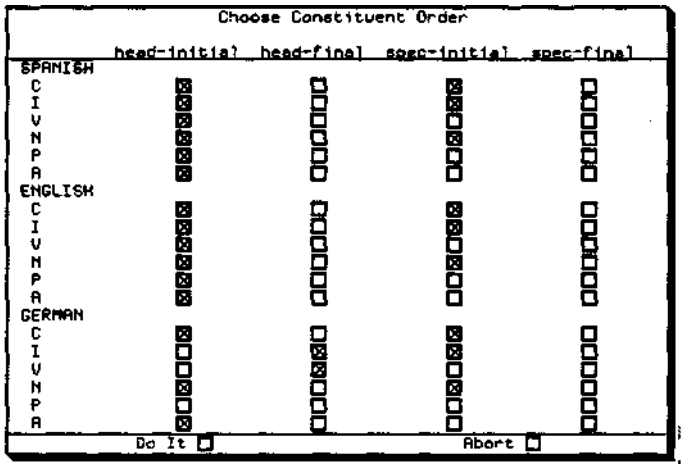

Figure 4: Choosing the Constituent Order Setting in the UNITRAN System

2 Design of the Syntactic Component of UNITRAN

One of the two major components of the UNITRAN system, the syntactic component, takes on the interlingual design shown in figure 3. This model of translation avoids the need for a detailed language-dependent specification of each source and target language. Instead, the source language is mapped into a form that is independent of any language, and the target-language form is then generated on the basis of parameter values for the selected language. Thus, there are no transfer modules or language-specific rules. The interlingual form is assumed to be a form common to all languages.

Note that the design of the UNITRAN translator allows the operation of the parser and generator to be modified without changing the programs underlying these modules. All of the principles associated with the system have user-modifiable parameter settings; thus, the source-language parser and target-language generator do not need to be modified (or replaced) when new languages are added. The only requirement is that the built-in parser and generator be *programmed* (via parameter settings) to process the source and target languages.

An example of setting a parameter in the UNITRAN system is shown in figure 4. Here, the user specifies the ordering of constituents with respect to a phrase for each language.[2] This parameter, called *constituent order,* is set to *head-initial* for English verbs, but *head-final* for verbs in many other languages including German and Japanese. The *head-initial* parameter setting forces the object to follow the verb in English (*e.g., hit the ball);* by contrast, the *head-final* parameter setting forces the object to precede the verb in German and Japanese (*e.g., the ball hit).* In addition to supplying the values for a small set of these parameters, the user must also provide a dictionary for each language, a necessary component in all machine translation systems.

The remainder of this paper will focus on the lexical-semantic module of UNITRAN showing how the lexical conceptual representation, coupled with the syntactic processing component described here, offers a solution to some of the syntactic and semantic divergences that arise during translation.

[2] This menu enumerates the syntactic categories for each language and allows the user to specify the constituent order associated with each category. Roughly, the spec-initial/spec-final setting corresponds to the positioning of subjects, and the head-initial/head-final setting corresponds to the positioning of objects.

3 Lexical Conceptual Structure in Machine Translation

This section describes the representation underlying the second major component of UNITRAN: the lexical semantic component. The work of Jackendoff (1983, 1990) has been the primary influence on the design of UNITRAN's lexical-semantic component. The representation adopted is *lexical conceptual structure* (henceforth LCS) as formulated by Hale &: Laughren (1983) and Hale & Keyser (1986). This representation has been adapted to the UNITRAN machine translation model in that it has been associated with an algorithm for recursive composition and decomposition of the interlingual form, and it has been linked systematically to the syntactic structure, both during parsing as well as during generation.

There are two fundamental properties of the adapted LCS representation that enable it to provide the basis for translation in difficult cases such as (1). The first is that the representation is *compositional* in nature. Thus, the representation underlying the *break-into* event allows the inherently compositional verb *break* to be chosen for the overtly compositional form *forzar la entrada* (literally, *force entry).* The second property of the LCS representation is that it provides an abstraction of lexical-semantic information from syntactic information. This abstraction allows the word *entrada* to be part of the lexical-semantic representation of the target-language sentence, even though this word is not syntactically realized (as it is in the source-language sentence). We will see in the following sections that compositionality and lexical-semantics/syntax abstraction are crucial to the model presented here. Before detailing the design of the LCS component, we will first look at the LCS representation and the mapping between this representation and the syntactic level of description.

3.1 The LCS Representation

UNITRAN requires a dictionary of lexical root-word entries for each language that is processed by the system. Each entry has two levels of description: the first is a lexical-semantic representation (the LCS of the lexical word), and the second is a mapping from the LCS representation to the syntactic structure (category and structural positioning of each argument associated with the lexical word). This section presents the LCS representation, and the next section describes the mapping from the LCS to the syntactic structure.

The best way to illustrate the form of the LCS is to present an example. The LCS that describes the *break-into* event is:

(3) **(CAUSE X (GO-LOC X (TO-LOC (IN-LOC X Y))) FORCEFULLY)**

This LCS description provides the meaning "**THING X** goes locationally into **THING Y** in a forceful manner." Figure 5 shows the underlying LCS tree structure generated from (3).

Figure 6 gives some examples of the lexical primitives used by the system. (Not all of the lexical primitives are listed here; see Dorr (1990).) In particular, I adopt Jackendoff's notions of **EVENT** and **STATE**; these are further specialized into such primitives as **CAUSE**, **GO**, **BE**, **STAY**, and **LET**. The specialized primitives are placed into Temporal, Locational, Possessional, Identificational, Circumstantial, Instrumental, Intentional, and Existential fields. For example, the primitive **GO-POSS** refers to a **GO** event in the Possessional field *(e.g.,* Beth received (= **GO-POSS**) the doll). If the **GO** event were placed in the Temporal field, it would become **GO-TEMP** *(e.g.,* the meeting went (= **GO-TEMP**) from 2:00 to 4:00). One difference between Jackendoff's representation and the one shown here is that the **POSITION**s (**AT-POSS**, **AT-LOC**, **WITH-INSTR**, *etc.)* are implemented as two-place predicates; thus, the X argument in figure 5 appears both internally and externally to the **IN-LOC** LCS node. Although the system uses only a small set of lexical-semantic primitives (approximately 25), this set

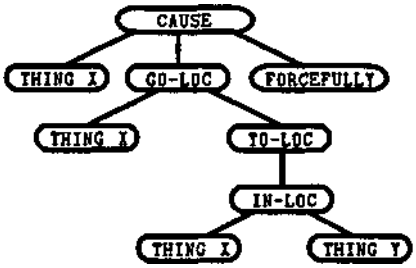

Figure 5: Underlying LCS For Break-into Event

LCS Type	*LCS Name*
EVENT	CAUSE, LET, GO-POSS, GO-IDENT, GO-TEMP,
	GO-LOC, STAY-POSS, STAY-TEMP
STATE	BE-IDENT, BE-POSS, BE-LOC, BE-TEMP
THING	ROOM, BOOK, PERSON, REFERENT
PROPERTY	TIRED, HUNGRY
PATH	TO-POSS, TO-LOC, FROM-POSS, FROM-LOC,
	TOWARD-POSS, TOWARD-LOC
POSITION	AT-POSS, AT-LOC, IN-LOC, ON-LOC, WITH-INSTR
LOCATION	HERE, THERE
TIME	TOMORROW, 9:00
MANNER	FORCEFULLY, WELL
INTENSIFIER	VERY

Figure 6: LCS Types and Names

is quite adequate for defining a potentially large corpus of words due to the compositional nature of LCS's. The advantage of a small set of primitives is that the search space is reduced during the lexical-selection stage of generation. The importance of a small search space will become more apparent later when we look at the lexical selection process in section 4.1.

3.2 Mapping From LCS to Syntactic Structure

In order to allow different target-language realizations of the *break-into* event, lexical entries must specify certain language-specific syntactic information. This is the nature of the LCS-to-syntax mapping associated with the definition of a word. The LCS-to-syntax mapping is incorporated into a word definition by means of three mechanisms. The first mechanism consists of two markers, :INT and :EXT, that map an LCS argument structure to a predicate-argument structure. A predicate-argument structure is an explicit syntactic representation of hierarchical relations between a predicate and its arguments. In particular, a predicate-argument structure embodies the asymmetry between the external argument position *(e.g.,* the subject of a verb) and the internal argument positions *(e.g.,* the objects of a verb). According to Rappaport &: Levin (1986), language-specific linking rules relate variables in an LCS to positions in a predicate-argument structure. For example, in English the *agent* argument is mapped to a position that is external to the predicate.

LCS Type	*Syntactic Category*
EVENT	V
STATE	V
THING	N
PROPERTY	A
PATH	P
POSITION	P
LOCATION	ADV
TIME	ADV
MANNER	ADV
INTENSIFIER	ADV

Figure 7: CSR Mapping from LCS Type to Syntactic Category

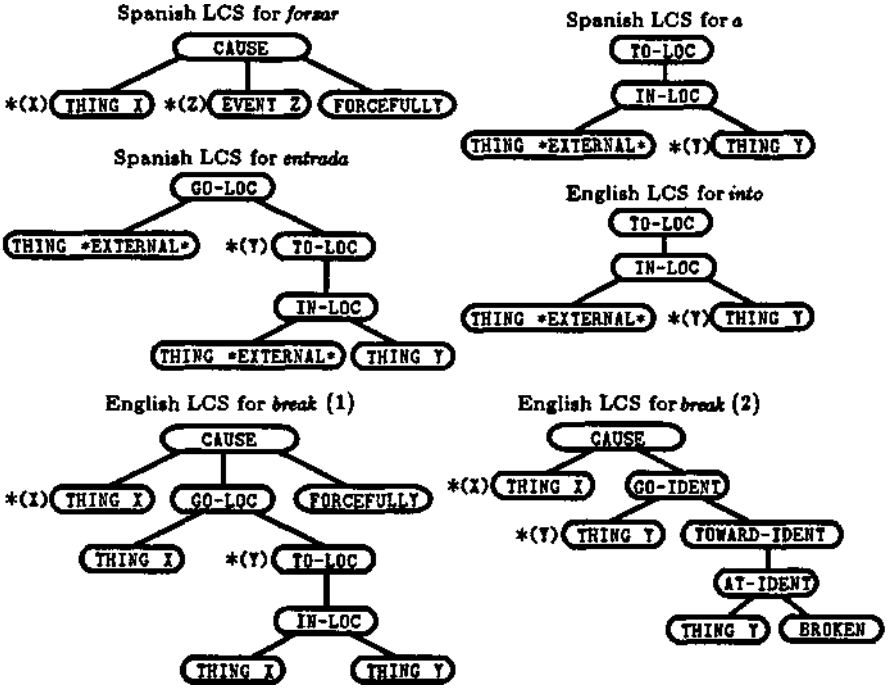

Figure 8: English and Spanish Lexical Entries for *forzar-break*

In UNITRAN, this process *is* implemented as a single, more general, language-independent linking routine that maps the hierarchically highest argument in the LCS to a syntactically external position, and all other arguments to syntactically internal positions. When this routine is to be overridden by a lexical entry, the language-specific markers :INT and :EXT are used.

The second LCS-to-syntax mechanism is the :CAT marker that provides a syntactic category for an LCS argument. According to Chomsky (1986), there is a language-specific function called CSR (canonical-syntactic representation) which provides a default mapping from the thematic-roles of GB theory to the appropriate syntactic categories. For example, in English the *agent* role is mapped to the category N. In UNITRAN, the CSR function has been implemented as a general language-independent routine that maps an LCS type to a syntactic category (see figure 7). When this mapping is to be overridden by a lexical entry, the language-specific marker :CAT is used.

The third LCS-to-syntax mechanism, developed specifically for the UNITRAN system, is the '*' marker; this marker provides a pointer to the position where arguments are explicitly realized in the surface form. Figure 8 shows how the '*' notation is used for the English and Spanish lexical entries that correspond to the *break-into* event. Note that the Spanish LCS for *forzar* contains a **Z** argument that must be filled in (as an **EVENT**), whereas there is no corresponding argument in the English LCS. This accounts for the distinction between the compound verb *forzar la entrada* and the single verb *break* in the translation example (1). The ***EXTERNAL*** symbol used in the LCS definitions of *a* and *entrada* is a place-holder for an LCS that will fill this position by means of lexical-semantic composition (to be described in the next section). For example, when the LCS associated with *entrada* is composed with the LCS associated with *forzar,* the **X** argument will replace the ***EXTERNAL*** place-holder in the LCS associated with *entrada.*

The '*' marker is required for every explicitly realized argument of a lexical entry, whereas the :INT, :EXT, and :CAT markers are used only in cases where the linking and CSR functions need to be overridden. Given this organization of syntactic and semantic information, a target-language syntactic structure can be generated from an underlying LCS structure. The design of the LCS component and the lexical selection and syntactic realization processes that allow syntactic and lexical-semantic decisions to be made will be described in the next section.

4 Design of the LCS Component in UNITRAN

Figure 9 shows the design of the LCS component of the system as described here and in Dorr (1990). Essentially, lexical-semantic processing involves three top-level tasks. The first is the mapping of each word to its corresponding LCS. The second is the composition of the resulting LCS forms into a single LCS that underlies the source- and target-language sentences. The third is the mapping of each node in the composed LCS to an appropriate target-language word, which is then projected to its phrasal (or *maximal)* level and attached according to the positioning requirements of the word that selects it.[3]

We return to our translation example shown in (1). The parsing module of the syntactic component supplies a source-language syntactic tree to the LCS component of the system. Figure 10 shows the source-language syntactic tree input for the current example.[4] When this syntactic tree is passed to the LCS component, an LCS is selected for each word (*Juan*, *forzar, entrada,* and *cuarto),* and a single underlying LCS is composed by means of the (reversible) LCS-to-syntax

[3] For discussion of projection to maximal level by the syntactic component of the system, see Dorr (1987). In a nutshell, **X-MAX** refers to the **XP** phrase that contains a word of category **X**.

[4] In this case, there is only one possible source-language tree; however, if the structure were ambiguous, other possibilities would be returned. The *e* elements under **C** and **I** are syntactic positions for which there is no overt lexical material.

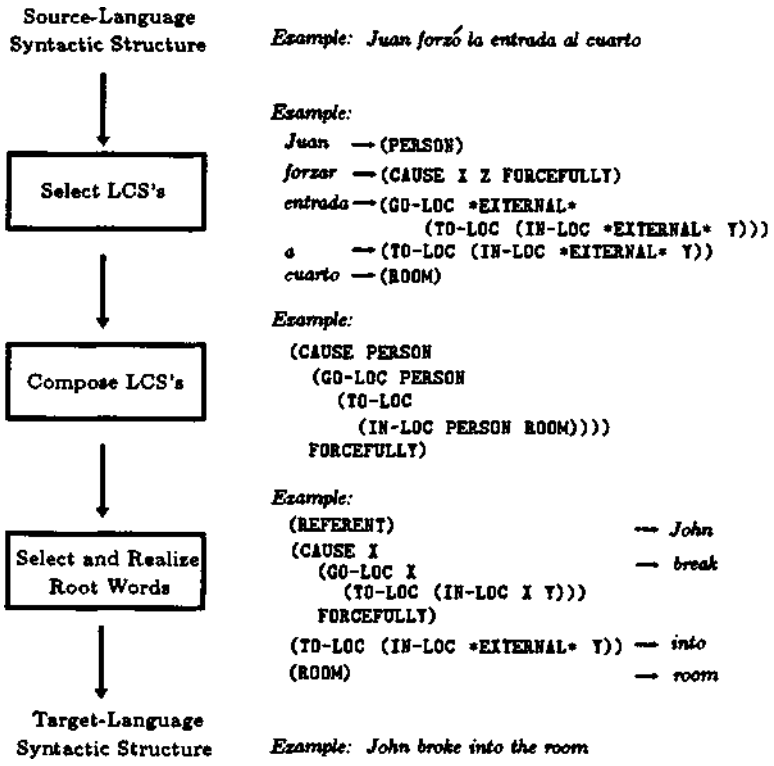

Figure 9: Design of the LCS Component in UNITRAN

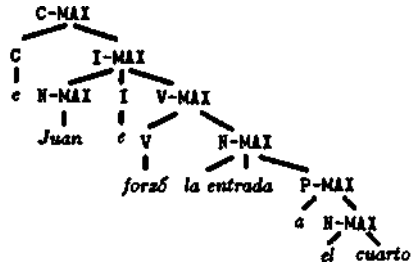

Figure 10: Source-Language Syntactic Tree for *Juan forzó la entrada al cuarto*

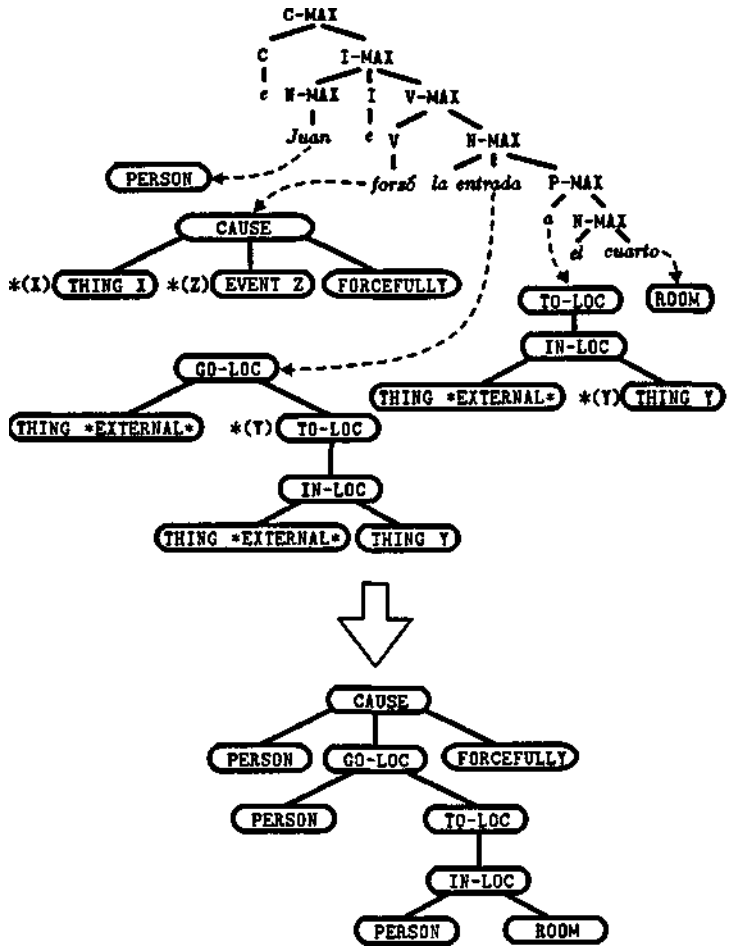

Figure 11: Mapping Syntactic Tree Positions to LCS Argument Positions: Derivation of the Composed LCS for the *Break-into* Event

mappings described in the last section. For example, the LCS **PERSON** is selected for the word *Juan* and the LCS **ROOM** is selected for the word *cuarto;* during LCS composition, the **PERSON** is mapped by the generalized linking routine to the external position **X** of *forzar,* and the **ROOM** is mapped by the same routine to the internal position **Y** of *entrada.*

Figure 11 shows the mapping from the syntactic tree to the LCS argument positions in the definition of *break* and *entrada;* the result of this mapping is the composed LCS as shown. Once the LCS has been composed, the third module of the LCS component performs *lexical selection* and *syntactic realization* to produce the final target-language tree and sentence. These two operations are the first and second steps of the procedure applied by this module of the LCS component as shown in figure 12.

Note that the third step, argument realization, is actually a recursive call to this procedure: arguments of a target-language word are realized in the same way that the target-language word was realized. We will now examine the lexical selection and syntactic realization steps in more detail, and we will see later (section 6) how these steps are applied to the current example.

1. Lexically select target-language word.
2. Syntactically realize and attach target-language word.
3. Realize arguments of target-language word, if any.

Figure 12: Procedure for Lexical Selection and Syntactic Realization

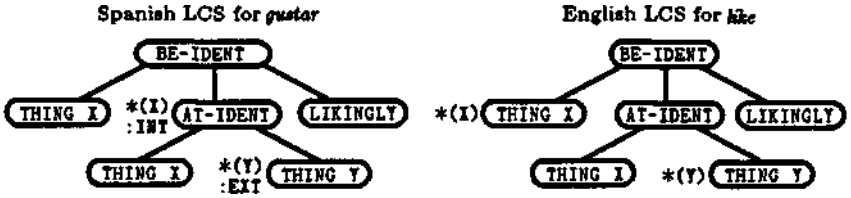

Figure 13: English and Spanish Lexical Entries for *gustar-like*

4.1 Lexical Selection: Thematic Divergence

This section describes the lexical selection task and provides an example showing how the LCS approach handles the problem of thematic divergence during this process. Lexical selection is the task of choosing the target-language words that accurately reflect the meaning of the corresponding source-language words. One of the difficulties of this task is the fact that the equivalent source- and target-language forms are potentially thematically divergent. An example of thematic divergence shows up in the translation of the Spanish word *gustar* to the English word *like.* Although these two verbs are semantically equivalent, their argument structures are not identical: the subject of *like (I)* is the *theme* of the action, whereas the subject of *gustar (María)* is the *agent* of the action. Thus, we have:

(4) Me gusta María (Mary pleases me) ⇒ I like Mary

In (4), the subject of the source-language sentence has freely inverted into post-verbal position. Thus, the post-verbal subject is considered to be the external argument of the main verb. Free subject-inversion is a property of *null subject* languages (i.e., languages such as Spanish, Italian, Hebrew, *etc.* as described in section 1); this property is taken into account during syntactic parsing and generation.

In a purely syntactic-based scheme, the semantics of the verb *gustar* would be lost because the literal translation *(to please)* would be selected for the target-language verb. In contrast, a semantic-based system would generally be able to make the correct lexical selection, but it might have difficulty with syntactic realization of the target-language arguments because it has no notion of syntactic argument divergence.

In the LCS approach, the underlying conceptual structure for *gustar* and *like* is identical. The only difference is that the syntactic mappings associated with these two verbs are language-specific. Figure 13 shows the LCS definitions underlying *gustar* and *like.* The LCS provides the meaning "**THING X** is in an identificational state **LIKINGLY** with respect to **THING Y**." However, the variables **X** and **Y** map to different syntactic positions for English and Spanish. The English version maps the **X** variable to the external position and the **Y** variable to the internal position, and the Spanish version interchanges these two positions by means of the :INT and :EXT markers. Thus, the agent of the

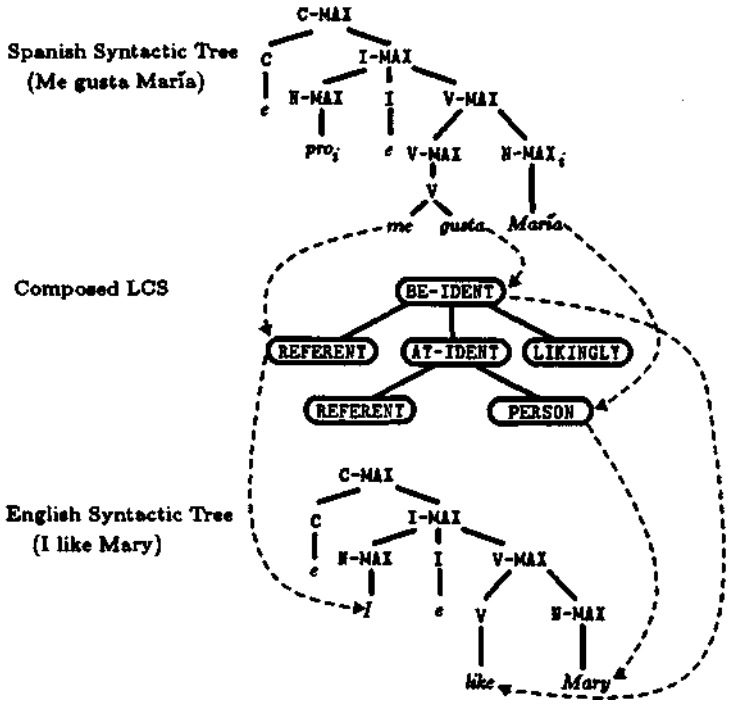

Figure 14: Translation of *Me gusta María* as *I like Mary*

action becomes the external argument (subject) in Spanish, and the internal argument (object) in English.

Lexical selection of a target-language word involves matching the composed LCS to the appropriate root word in a target-language possibility set. For example, suppose the system is trying to select the appropriate target-language token for the composed LCS that corresponds to the source-language verb *gustar.* Several target words (including *like, sleep,* and many others that use the **BE-IDENT** LCS) are selected as possible lexical candidates. Each of these candidates is then examined for a match: not only must the top-level LCS coincide, but all LCS's under the top-level LCS must also coincide. In general, there are two classes of LCS nodes that are taken into consideration during the matching process of lexical selection. The more general nodes (*e.g.,* **BE-IDENT**, **GO-POSS**, *etc.)* allow the matcher to determine the LCS class of the target-language term; the more specific nodes *(e.g.,* **LIKINGLY**, **FORCEFULLY**, *etc.)* are used for final convergence on a particular target-language term such as *like* as opposed to *love,* and *force* as opposed to *cause.* Because the search space is reduced *(i.e.,* the number of primitives is small), the final convergence on a target-language term is not as costly as it would be otherwise.

In the current example, the system determines that the *like* LCS is a match because it contains a **BE-IDENT** event whose arguments coincide with the arguments of the **BE-IDENT** in the composed LCS. Figure 14 shows the mapping from the source-language syntactic tree to the target-language syntactic tree by means of the composed LCS for example (4).

Note that, even though the arguments are not syntactically realized in the same way, the lexical selection procedure still succeeds. This is because of the separation between the syntactic description and the conceptual description. LCS descriptions provide the abstraction necessary for lexical selection without regard to syntax. This abstraction is an advance over other approaches because it

provides an accurate translation of the source-language terms despite thematic divergences. In section 6, we will see how the LCS-to-syntax mappings provide the necessary mechanism for syntactic realization without regard to conceptual considerations.

4.2 Syntactic Realization: Conflational Divergence

Syntactic realization is the second step applied by the third module of the LCS component. This step involves mapping an LCS to a syntactic representation using the LCS-to-syntax mapping described previously. A problem associated with this task is that source- and target-language terms have potentially divergent argument incorporation (or *conflation)* characteristics. According to Talmy (1983), verbs may have a semantic representation that is not entirely exhibited at the level of syntactic structure. For example, the verb *enter* incorporates a *conflated* or "understood" particle *into* as part of its meaning structure; this particle manifests itself in the similar composite predicate *break into.*

As it turns out, the Spanish equivalent of *break into (forzar)* has an additional conflated argument *entrada* (literally, *entry);* this argument is "understood," but not syntactically realized in English. We will return to the forzar-break example in section 6, and we will see how this conflational divergence problem is solved. Now that we have looked at the mechanisms involved in solving divergences in UNITRAN, we will look at how other approaches have attempted to solve this problem.

5 Related Work

In tackling the more global problem of machine translation, many researchers have addressed different pieces of the divergences described here, but no single approach has yet attempted to solve all of the divergence types handled by UNITRAN. (In addition to those already mentioned, we will look briefly at some of the others in section 7.) Furthermore, the pieces that *have* been solved are accounted for by mechanisms that are not general enough to carry over to other pieces of the problem, nor do they take advantage of cross-linguistic uniformities that can tie seemingly different languages together.

The LCS representation has commonly been compared to the conceptual dependency (CD) representation (see Schank (1974), and Lytinen & Schank (1982)). In general, the approaches that use the CD representation are similar to that of UNITRAN in that they use the representation as the interlingual form underlying the source- and target-language sentence. Furthermore, the representation relies on a set of *primitives* that serve as the basic units of meaning. However, the traditional Schank-style semantic primitives differ from those of UNITRAN in a number of ways. In particular, the Jackendoff-style primitives are not intended to be *the* building blocks for natural language; it is expected that the set of primitives will be extended (and, in fact, it *has* already been extended to include verbs of communication and perception). In contrast, the primitives used for Schank-style systems are limited to a small handful (approximately 14), which are intended to be combined in various ways by a number of different operators. In such a system, it is not clear how to distinguish, for example, between verbs like *want, know, think, believe, etc.,* which rely solely on the **MTRANS** primitive.

Another key distinction between the LCS approach and the CD approach is that the latter lacks a generalized linking to syntax. For example, there is no systematic method for determining which conceptual argument of a CD representation is the subject and which is the object. This means that there is no uniform mechanism for handling divergences such as the subject-object reversal

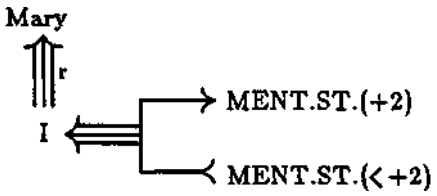

Figure 15: CD Representation for *I like Mary*

of *I like Mary* ⇒ *Me gusta María.* Figure 15 shows the CD representation for this sentence.[5] In the CD approach, there is no single representation that is common to both the English and Spanish versions of this sentence. Furthermore, no divergence mechanism has been proposed in the CD framework; thus, there is no way to capture the subject-object reversal of this example. By contrast, in the LCS approach, the :INT and :EXT markers mentioned in section 3.2 are used for this type of divergence. As we saw in figure 14, the LCS representation that is analogous to this CD representation for this example is used both for the Spanish sentence as well as for the English sentence. However, as shown in figure 13, the Spanish lexical entry for *gustar* associates the :INT marker with the logical subject (**REFERENT** in this example) and the :EXT marker with the non-subject argument (**PERSON** in this example), thus accounting for the thematic divergence.

The LMT system is a logic-based English-German machine translator based on a modular logical grammar (McCord, (1989)). McCord specifically addresses the problem of thematic divergence in translating the sentence *Mir gefällt der Wagen (I like the car)*. However, the solution that is offered is to provide a "transfer entry" that interchanges the subject and object positions. There are two problems with this approach. First it relies specifically on this object-initial ordering, even though the sentence is arguably more preferable with a subject-initial ordering *Der Wagen gefällt mir;* thus, the solution is dependent on syntactic ordering considerations, and will not work in the general case. Second the approach does not attempt to tie this particular type of thematic divergence to the rest of the space of thematic divergence possibilities; thus, it cannot uniformly translate a conceptually similar sentence *Ich fahre das Wagen gern (I like to drive the car).*

With respect to the task of lexical selection, a number of other approaches are relevant. In particular, the LCS matching process that determines the appropriate target-language word is related to that of Miezitis (1988) in that both approaches attempt to find all correct and usable matches rather than the "best" match (as in the preference semantics approach of Wilks (1973)). Several researchers have identified problems with taking such an open-ended approach to generation. In particular, the issue of efficiency is addressed by Jacobs (1985) with respect to the discrimination net approach to generation of Goldman (1974). This approach to verb selection is similar to that of UNITRAN in that the target-language verbs are narrowed down according to selectional restrictions associated with argument positions. In Goldman's generator, each primitive of the system could potentially lead to a wide range of verbs, depending on the results of the tests in the discrimination nets. The UNITRAN system provides a richer set of primitive units, but, as Jacobs mentions, such a system is subject to a proliferation of primitives due to the need to distinguish among a number of different predicates. The lexical selection problem has been addressed by Pustejovsky & Nirenburg (1987) and Nirenburg & Nirenburg (1988) with respect to the DIOGENES generation system. This system uses a constrained discrimination network approach for the selection of open-class lexical items, and it uses discourse information in order to select the appropriate lexicalizations

[5] See Schank (1974) for a description of the notation and additional examples.

for closed-class lexical items. This investigation has illustrated the importance of discourse and focus information in the generation process; such information has not yet been incorporated into the UNITRAN system, but could prove useful in future versions.

6 Forzar-Break Revisited

We now return to our translation example: *Juan forzó la entrada al cuarto.* The argument structures are shown in (5):

(5)
Juan forzó la entrada al cuarto (John forced entry to the room)
[$_{V.MAX}$ [$_{V}$ forzar] [$_{N.MAX}$ la entrada] [$_{P.MAX}$ a • • •]]
⇓
John broke into the room
[$_{V.MAX}$ [$_{V}$ break] [$_{P.MAX}$ into • • •]]

Having described the lexical selection and syntactic realization processes, we can complete this example. Refer to figure 8 for the LCS definitions underlying the English and Spanish tokens in this sentence. Once the LCS for this sentence has been composed (see figure 11), the lexical selection procedure (step 1 of figure 12) must choose the appropriate English root words, and the syntactic realization procedure must syntactically produce the appropriate English structures.

There are three difficult tasks in the translation of *forzar* to *break:* selection of the predicate *break,* suppression (conflation) of the *entry* argument, and realization of the particle *into.* A syntactic-based scheme has no notion of compositionality and would fail immediately in trying to map *forzar* (literally *force)* to *break* (or *vice versa).* Furthermore, it would have the problem of choosing the appropriate particle, even if it were able to provide the correct structure (a prepositional phrase). On the other hand, a robust semantic-based scheme would have the ability to compose *forzar* and *entrada,* but it would not be able to determine whether the target-language argument was to be left implicit or whether it was to be syntactically realized, because there is no notion of conflation in such a scheme.

The LCS scheme uses compositionality to map *forzar la entrada* to *break:* the LCS for *forzar* contains a **CAUSE**, and the LCS for *entrada* contains a **GO-LOC**, both of which combine to match the composite LCS for *break.* As we can see from the definitions in figure 8, the LCS for *forzar* contains the **CAUSE** portion of the *break-into* action, and the LCS for *entrada* contains the locational part of the *break-into* action.

Note that there is another LCS for the word *break* (2) that corresponds to "breaking an object." For the *break into* example, the matching routine of the lexical selection procedure succeeds on the first LCS definition because it is a **GO-LOC** and (correctly) fails on the second LCS definition because it is a **GO-IDENT**. At this point, the syntactic realization procedure determines that the **GO-LOC** LCS is not overtly realized for the predicate *break* because it is not associated with a '*' marker. Thus, this argument is left unrealized (fulfilling the conflation task). However, the **TO** argument is marked with a '*' in the LCS-to-syntax mappings of *break,* so the system matches this argument with the **TO** LCS of *into,* and the phrase *into the room* is realized. Figure 16 shows the entire mapping from the source-language syntactic tree to the target-language syntactic tree by means of the composed LCS for example (5). The final output sentence is produced by reading off the leaves of the target-language syntactic tree.

Note that the lexical selection and syntactic realization procedures succeed even though there are lexical and conflational differences between the source- and target-language sentences. This is because the LCS-to-syntax mapping and the compositional nature of LCS's allow syntactic distinctions for conceptually equivalent forms. LCS-to-syntax mappings and compositionality provide

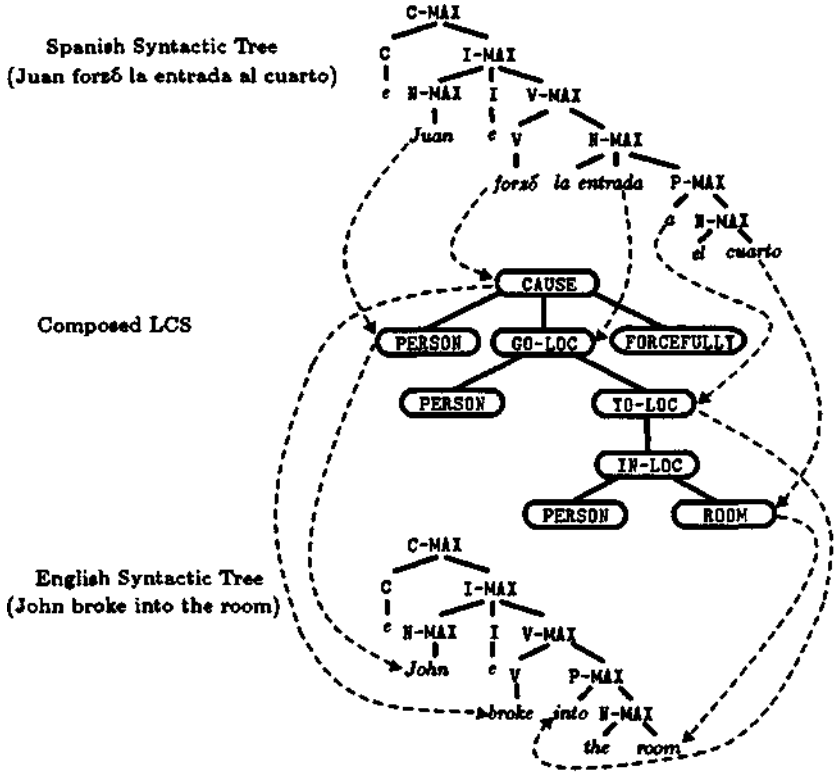

Figure 16: Translation of *Juan forzó la entrada al cuarto* as *John broke into the room*

an advantage to the LCS approach compared to other approaches because they allow structural and conflational divergences to be accounted for during the syntactic realization portion of the translation process.

7 Other Examples

In addition to the divergence types already mentioned (thematic and conflational), there are a number of other divergence types that are handled by UNITRAN. Figure 17 shows a subset of these divergence types with respect to English, Spanish, and German.[6]

We will look at each of these traditionally difficult divergence types in turn. The first divergence type is a structural divergence in that the verbal object is realized as a noun phrase *(John)* in English and as a prepositional phrase (a Juan) in Spanish. The second example exhibits conflational divergence as we have already seen: the Spanish equivalent of *break into (forzar)* has an additional conflated argument *entrada* (literally, *entry)* that is "understood," but not syntactically realized in English. The third divergence type is a lexical divergence as illustrated by the choice of a different lexical word *haben* (literally *have)* in German for the English word *like.* The fourth divergence type is categorial in that the predicate is adjectival *(hungry)* in English but nominal *(Hunger)* in German. Finally, the fifth divergence type is the thematic divergence observed in section 4.1: the object *(Mary)* of the English sentence is translated as the subject *(María)* in the Spanish sentence.

[6] Many sentences may fit into these divergence classes, not just the ones listed here. Also, a single sentence may exhibit any or all of these divergences.

Divergence Type	*Translation Example*
Structural	I saw John ⇳ Vi a Juan (I saw to John)
Conflational	John broke into the room ⇳ Juan forzó la entrada al cuarto (John forced entry to the room)
Lexical	I like Mary ⇳ Ich habe Marie gern (I have Mary likingly)
Categorial	I am hungry ⇳ Ich habe Hunger (I have hunger)
Thematic	I like Mary ⇳ Me gusta María (Mary pleases me)

Figure 17: Divergence Types in Machine Translation

Because of space limitations, we will not describe the translation process for each of these examples, but the reader is referred to Dorr (1990).

8 Limitations and Future Work

UNITRAN was deliberately designed to operate on one sentence at a time, and as such there are a number of inherent limitations. For example, the system does not incorporate context or domain knowledge; thus, it cannot use discourse, situational expectations, or domain information in order to generate a sentence. Consequently, there are a number of capabilities found in other systems that cannot be reproduced here including external pronominal reference (as in MUMBLE, McDonald (1983, 1987)), paraphrasing (as in MOPTRANS, Lytinen & Schank (1982)), story telling (as in SAM, Schank & Abelson (1977) and Cullingford (1986)), interactive question-answering (as in TEXT, McKeown (1985)), *etc.* This is not to say that issues of context and domain knowledge should be ignored; on the contrary, these types of knowledge may be the next step in the evolution of the UNITRAN system.

An additional limitation of the LCS approach is the potential for generating several target-language possibilities for a given lexical-semantic primitive. It is possible that the LCS-matching procedure will not adequately cut down the target-language possibilities during the mapping from LCS to lexical items. For example, there are many open-ended classes of words (in particular, proper and common nouns, and certain adjectives and adverbs) that are not distinguishable by their LCS's. This is because LCS's provide an underlying representation of predicates and their arguments; any lexical item that does not exhibit a predicate-argument relationship must be translated by other means. Thus, if the possibility list is still quite large (more than two or three lexical items) after

LCS-matching routines have finished the lexical selection process, a direct-mapping routine is used instead for lexicalization. That is, certain lexical-items *(John, book, red, quickly, etc.)* may be selected on the basis of a direct mapping to the surface form. As mentioned in section 5, an approach to generation of lexical items based on focus information is presented in Pustejovsky & Nirenburg (1987) and Nirenburg & Nirenburg (1988). Because the system described here does not include a model of discourse, the direct-mapping technique is used for such problematic cases.

Another limitation of the system as it stands is that the notion of *aspect* is not represented in the LCS structures. For example, there is no way to establish whether an event is prolonged, repeated, instantaneous, *etc.* Thus, in the sentence *I stabbed John,* there is no way to determine how many times John was stabbed. As it turns out, the Spanish surface realization relies on this missing information. The translation of the repetitive version of *stab* is the surface form *dar puñaladas* (the plural form of knife-wound), whereas the translation of the non-repetitive version is the surface form *dar una puñalada* (the singular form of knife-wound). Jackendoff *does* try to include the notion of aspect in some cases. For example, the lexical-semantic token **BE-CIRC** allows the progressive aspect to be expressed. However, there is no way to determine the appropriate aspect in the general case, so the system arbitrarily chooses a target-language word when such an ambiguity arises. Superimposing a system of aspect (see, for example, Tenny (1989) and Brent (forthcoming)) could prove to be useful in the future.

9 Summary

The UNITRAN system is implemented in Common Lisp and is currently running on a Symbolics 3600 series machine. The syntactic component of the system is based on GB theory, and the lexical-semantic component of this system is based on LCS theory. The principles-and-parameters approach provided by the GB-based syntactic component of the system has been shown to be valuable for machine translation because it accounts for several types of surface-syntactic phenomena across diverse languages. The LCS approach has been shown to be valuable for machine translation because it facilitates two crucial translation operations: lexical selection and syntactic realization. Furthermore, because it is compositional in nature, the LCS representation aids in tackling the difficult problems of structural, thematic, and conflational divergence.

We have seen that the definition of a potentially large (theoretically infinite) set of words is supported by the ability to combine the same lexical-semantic primitives in an indefinite number of ways. However, the search space for root-word selection is not explosive because there are only a small number of primitives that must be searched at each level of the matching procedure. In addition, LCS descriptions seem to provide the abstraction necessary for selecting appropriate target-language terms with minimal dependence on syntax, and they also provide the necessary mechanism for realizing arguments without regard to conceptual considerations. In particular, lexical entries are divided into two levels of description—lexical-semantic and syntactic—the former used for lexically selecting arguments, and the latter used for syntactically "fitting" these arguments into a predicate-argument structure.

In its description of the shift toward interlingual machine translation, this paper has demonstrated a need for a translator to operate cross-linguistically despite the potential idiosyncrasies for a given language. Two types of knowledge (language-specific and language-independent) have been shown to be crucial for fulfillment of the translation task. Both of these types of knowledge have proven to be useful during syntactic processing and semantic processing.

The approach presented here tries to incorporate some of the more promising syntactic and semantic aspects of existing translation systems. Specifically, the model incorporates structural

information for realization and positioning of arguments. Unlike direct-replacement and entirely syntactic-based approaches, however, it avoids non-compositional direct-mapping word selection. In addition, the model has the ability to select target terms on the basis of compositional properties. Unlike many semantic-based approaches, however, it does not rely upon context-dependent routines, and it does not entirely abandon syntactic considerations for selection and realization of root words and their associated arguments.

In summary, this paper has shown that the UNITRAN system solves a number of traditional problems for machine translation by combining:

- A principles-and-parameters-based component to handle syntactic variation, and
- An LCS-based component to handle problems of thematic, structural, and conflational divergence.

10 Acknowledgements

This paper describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for this research has been provided by NSF Grant DCR-85552543 under a Presidential Young Investigator's Award to Professor Robert C. Berwick. Useful guidance and commentary during this research were provided by Bob Berwick, Noam Chomsky, Bruce Dawson, Ken Hale, Mike Kashket, Jeff Siskind, and Patrick Winston. The author is also indebted to two anonymous reviewers for their aid in reshaping this paper into its current form.

11 References

Brent, Michael R. (forthcoming) "Automating Lexicography, *or* Syntax of the Theory of Aspects," Ph.D. thesis, Massachusetts Institute of Technology.

Chomsky, Noam A. (1981) *Lectures on Government and Binding,* Foris Publications, Dordrecht.

Chomsky, Noam A. (1982) "Some Concepts and Consequences of the Theory of Government and Binding," MIT Press.

Chomsky, Noam A. (1986) *Knowledge of Language: Its Nature, Origin and Use,* MIT Press, Cambridge, MA.

Cullingford, Richard E. (1986) *Natural Language Processing: A Knowledge-Engineering Approach,* Rowman and Littlefield, Totowa, New Jersey.

Dorr, Bonnie J. (1987) "UNITRAN: A Principle-Based Approach to Machine Translation," AI Technical Report 1000, Master of Science thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Dorr, Bonnie J. (1989a) "Lexical Conceptual Structure and Generation in Machine Translation," *AI Memo 1160, Proceedings of the Eleventh Annual Conference of the Cognitive Science Society,* Ann Arbor, Michigan.

Dorr, Bonnie J. (1989b) "Conceptual Basis of the Lexicon in Machine Translation," *AI Memo 1166, First Annual Workshop on Lexical Acquisition, IJCAI-89,* Detroit, Michigan.

Dorr, Bonnie J. (1990) "Lexical Conceptual Structure and Machine Translation," Ph.D. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Goldman, Neil M. (1974) "Computer Generation of Natural Language from a Deep Conceptual Base," Ph.D thesis, Department of Computer Science Department, Stanford University, Stanford, CA.

Hale, Kenneth and M. Laughren (1983) "Warlpiri Lexicon Project: Warlpiri Dictionary Entries," Massachusetts Institute of Technology, Cambridge, MA, Warlpiri Lexicon Project.

Hale, Kenneth and Jay Keyser (1986) "Some Transitivity Alternations in English," Center for Cognitive Science, Massachusetts Institute of Technology, Cambridge, MA, Lexicon Project Working Papers #7.

Jackendoff, Ray S. (1983) *Semantics and Cognition,* MIT Press, Cambridge, MA.

Jackendoff, Ray S. (1990) *Semantic Structures,* MIT Press, Cambridge, MA.

Jacobs, Paul Shafran (1985) "A Knowledge-Based Approach to Language Production," Ph.D. thesis, University of California.

Lytinen, Steven and Roger Schank (1982) "Representation and Translation," Department of Computer Science, Yale University, New Haven, CT, Technical Report 234.

McCord, Michael C. (1989) "Design of LMT: A Prolog-Based Machine Translation System," *Computational Linguistics* 15:1, 33-52.

McDonald, David D. (1983) "Natural Language Generation as a Computational Problem," in *Computational Models of Discourse,* Brady, Michael and Robert C. Berwick (eds.), MIT Press, Cambridge, MA, 209-265.

McDonald, David D. (1987) "Natural Language Generation: Complexities and Techniques," in *Machine Translation: Theoretical and Methodological Issues,* Sergei Nirenburg (ed.), Cambridge University Press, Cambridge, England.

McKeown, Kathleen (1985) *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text,* Cambridge University Press, Cambridge, England.

Miezitis, Mara (1988) "Generating Lexical Options by Matching in a Knowledge Base," Technical Report CSRI-217, Master of Science thesis, Department of Department of Computer Science, University of Toronto.

Nirenburg, Sergei and Irene Nirenburg (1988) "A Framework for Lexical Selection in Natural Language Generation," *Proceedings of COLING-88,* Budapest, Hungary, 471-475.

Pustejovsky, James and Sergei Nirenburg (1987) "Lexical Selection in the Process of Language Generation," *Proceedings of the 25th Annual Conference of the Association for Computational Linguistics,* Stanford University, Stanford, CA, 201-206.

Rappaport, Malka and Beth Levin (1986) "What to Do with Theta-Roles," Center for Cognitive Science, Massachusetts Institute of Technology, Cambridge, MA, Lexicon Project Working Papers #11.

Schank, Roger (1974) *Conceptual Information Processing,* Elsevier Science Publishers, Amsterdam, The Netherlands.

Schank, Roger C. and Robert Abelson (1977) *Scripts, Plans, Goals, and Understanding,* Lawrence Erlbaum Associates, Inc., Hillsdale, NJ.

Talmy, Leonard (1983) "How Language Structures Space," in *Spatial Orientation: Theory, Research, and Application,* Pick, Herbert L., Jr., and Linda P. Acredolo (eds.), Plenum Press, New York.

Tenny, Carol (1989) "The Aspectual Interface Hypothesis," Center for Cognitive Science, Massachusetts Institute of Technology, Cambridge, MA, Lexicon Project Working Papers #31.

Wilks, Yorick (1973) "An Artificial Intelligence Approach to Machine Translation," in *Computer Models of Thought and Language,* R. C. Schank and K. M. Colby (eds.), Freeman, San Francisco, CA, 114-151.