

# Large-Scale Acquisition of LCS-Based Lexicons for Foreign Language Tutoring

Bonnie J. Dorr

Department of Computer Science  
University of Maryland  
College Park, MD 20742, USA  
bonnie@cs.umd.edu

## Abstract

We focus on the problem of building large repositories of *lexical conceptual structure* (LCS) representations for verbs in multiple languages. One of the main results of this work is the definition of a relation between broad semantic classes and LCS meaning components. Our acquisition program—LEXICALL—takes, as input, the result of previous work on verb classification and thematic grid tagging, and outputs LCS representations for different languages. These representations have been ported into English, Arabic and Spanish lexicons, each containing approximately 9000 verbs. We are currently using these lexicons in an operational foreign language tutoring and machine translation.

## 1 Introduction

A wide range of new capabilities in NLP applications such as foreign language tutoring (FLT) has been made possible by recent advances in lexical semantics (Carrier and Randall, 1993; Dowty, 1991; Fillmore, 1968; Foley and Van Valin, 1984; Grimshaw, 1990; Gruber, 1965; Hale and Keyser, 1993; Jackendoff, 1983; Jackendoff, 1990; Jackendoff, 1996; Levin, 1993; Levin and Rappaport Hovav, To appear; Pesetsky, 1982; Pinker, 1989). Many of these researchers adopt the hypothesis that verbs can be grouped into broad classes, each of which corresponds to some combination of basic meaning components. This is the basic premise underlying our approach to multilingual lexicon construction. In particular, we have organized verbs into broad semantic classes and subsequently designed a set of *lexical conceptual structures* (LCS), for each class. These representations have been ported into English, Arabic, and Spanish lexicons, each containing approximately 9000 verbs.

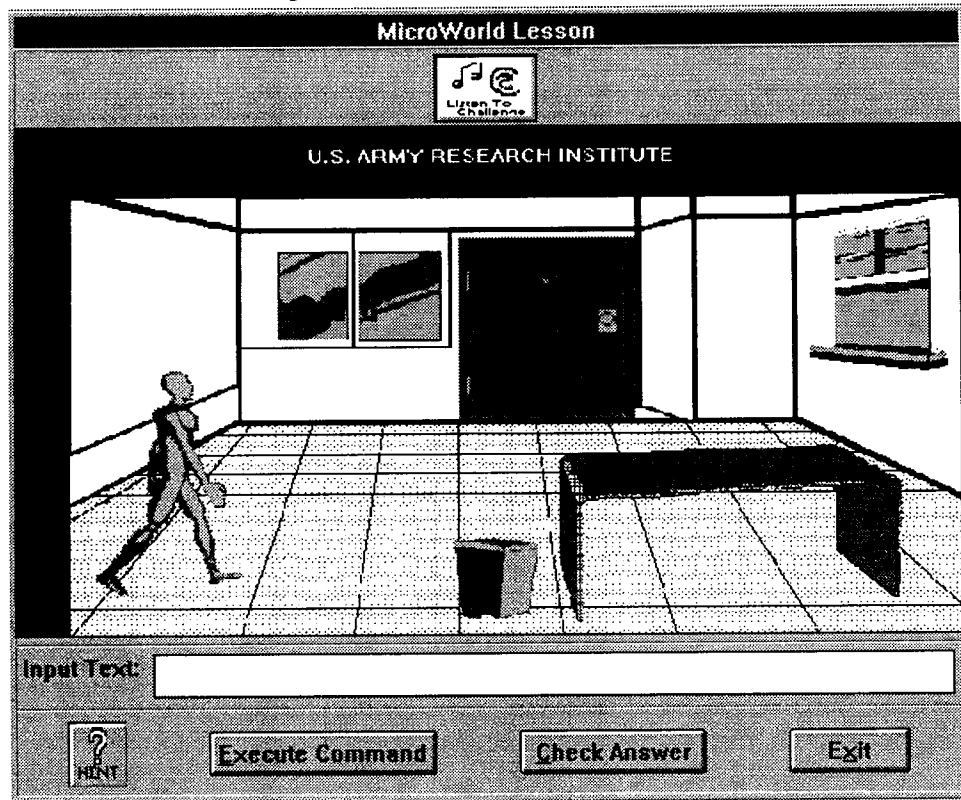
An example of a NLP application for which these lexicons are currently in use is an operational foreign

language tutoring (FLT) system called Military Language Tutor (MILT). This system provides a wide range of lessons for use in language training. One of the tutoring lessons, the MicroWorld Lesson (see Figure 1) requires the capability of the language-learner to state domain-specific actions in a variety of different ways. For example, the language-learner might command the agent (pictured at the left in the graphical interface) to take the following action: *Walk to the table and pick up the document*. The same action should be taken if the user says: *Go to the table and remove document*, *Retrieve the document from the table*, etc. The LCS representation provides the capability to execute various forms of the same command without hardcoding them as part of the graphical interface.

In another tutoring lesson, Question-Answering, the student is asked to answer questions about a foreign language text that they have read. Their answer is converted into an LCS which is matched against a prestored LCS corresponding to an answer typed in by a human instructor (henceforth, called the “author”). The prestored LCS is an idealized form of the answer to a question, which can take one of many forms. Suppose, for example, the question posed to the user is: *Where did Jack put the book?* (or *Adónde puso Jack el libro?* in Spanish). The author’s answer, e.g., *Jack put the book in the trash*, has been stored as an LCS by the tutoring system. If the student types *Jack threw the book in the trash*, or *Jack moved the book from the table into the trash*, the system is able to match against the prestored LCS and determine that all three of these responses are semantically appropriate.

We have developed an acquisition program—LEXICALL—that allows us to construct LCS-based lexicons for the FLT system. This program is designed to be used for multiple languages, and also for other NLP applications (e.g., machine translation). One of the main results of this work is the definition of a relation between broad semantic classes (based on work by Levin (1993)) and LCS meaning components. We build on previous work, where verbs were classified automatically (Dorr and Jones, 1996;

Figure 1: MicroWorld Lesson in MILT



Dorr, To appear) and tagged with thematic grid information (Dorr, Garman, and Weinberg, 1995). We use these pre-assigned classes and thematic grids as input to LEXICALL. The output is a set of LCS's corresponding to individual verb entries in our lexicon.

Previous research in automatic acquisition focuses primarily on the use of statistical techniques, such as bilingual alignment (Church and Hanks, 1990; Klavans and Tzoukermann, 1995; Wu and Xia, 1995) or extraction of syntactic constructions from on-line dictionaries and corpora (Brent, 1993). Others have taken a more knowledge-based (interlingual) approach (Lonsdale, Mitamura, and Nyberg, 1995). Still others (Copestake et al., 1995), use English-based grammatical codes for acquisition of lexical representations.

Our approach differs from these in that it exploits certain linguistic constraints that govern the relation between a word's surface behavior and its corresponding semantic class. We demonstrate that—by assigning a LCS representation to each semantic class—we can produce verb entries on a broad scale; these, in turn, are ported into multiple languages. We first show how the LCS is used in a FLT system. We then present an overview of the LCS acquisition process. Finally, we describe how LEXICALL constructs entries for specific lexical items.

## 2 Application of the LCS Representation to FLT

One of the types of knowledge that must be captured in FLT is linguistic knowledge at the level of the lexicon, which covers a wide range of information types such as verbal subcategorization for events (e.g., that a transitive verb such as *hit* occurs with an object noun phrase), featural information (e.g., that the direct object of a verb such as *frighten* is animate), thematic information (e.g., that *Mary* is the agent in *Mary hit the ball*), and lexical-semantic information (e.g., spatial verbs such as *throw* are conceptually distinct from verbs of possession such as *give*). By modularizing the lexicon, we treat each information type separately, thus allowing us to vary the degree of dependence on each level so that we can address the question of how much knowledge is necessary for the success of the particular NLP application.

This section describes the use of the LCS representation in a question-answering component of the MILT system (Sams, 1993; Weinberg et al., 1995). As described above, the LCS representation is used as the basis of matching routines for assessing students' answers to free response questions about a short foreign language passage. In order to inform the student whether a question has been answered

Table 1: Correspondence Between NLP Output and Tutor Feedback

System Prompt: Where did Jack put the book?			
Student Answer	Prestored Answer	Matcher Output	Feedback
Jack threw the book in the trash	Jack threw the book in the trash	exact match	"That's right"
Jack put the book in the trash	Jack threw the book in the trash	missing MANNER	"How?"
Jack threw the book in the trash	Jack put the book in the trash	extra MANNER	"You're assuming things"
Jack is friendly	Jack put the book in the trash	mismatch primitive	"Please reread"
Jack threw the book	Jack put the book in the trash	missing argument	"Where?"

correctly, the author of the lesson must provide the desired response in advance. The system parses and semantically analyzes the author's response into a corresponding LCS representation which is then pre-stored in a database of possible responses. Once the question answering lesson is activated, each of the student's responses is parsed and semantically analyzed into a LCS representation which is checked for a match against the corresponding pre-stored LCS representation. The student is then informed as to whether the question has been answered correctly depending on how closely the student's response LCS matches the author's pre-stored LCS.

Consider what happens in a lesson if the author has specified that a correct answer to the question *Adónde puso Jack el libro?* in Spanish is *Jack tiró el libro a la basura* ('Jack threw out the book into the trash'). This answer is processed by the system to produce the following LCS:

```
(1) [Event CAUSE
    ([Thing JACK],
    [Event GOLoc
    ([Thing BOOK],
    [Path TOLoc
    ([Position ATLoc
    ([Thing BOOK], [Thing TRASH])])])])])
    [Manner THROWINGLY]]
```

The LCS is stored by the tutor and then later matched against the student's answer. If the student types *Jack movio' el libro de la mesa a la basura* ('Jack moved the book from the table to the trash'), the system must determine if these two match. The student's sentence is processed and the following LCS structure is produced:

```
(2) [Event CAUSE
    ([Thing JACK],
    [Event GOLoc
    ([Thing BOOK],
    [Path TOLoc ([Position ATLoc
    ([Thing BOOK], [Thing TRASH])])])])])
    [Path FROMLoc ([Position ATLoc
    ([Thing BOOK], [Thing TABLE])])])])
```

The matcher compares these two, and produces the following output:

```
Missing: MANNER THROWINGLY
Extra: FROM LOC
```

The system identifies the student's response as a match with the pre-stored answer, but it also recognizes that there is one piece of missing information and one piece of extra information.

The "Missing" and "Extra" output is internal to the NLP component of the Tutor, i.e., this is not the final response displayed to the student. The system must convert this information into meaningful feedback so that the student knows how to repair the answer that was originally given. For example, the instructor can program the tutor to notify the student about the omitted information in the form of a 'How' question, or it can choose to ignore it. The extra information is generally ignored, although it is recorded in case the instructor decides to program the system to notify the student about this as well. The full range of feedback is not presented here. Some possibilities are summarized (in English) in Table 1 (adapted from (Holland, 1994)). Note that the main advantage of using the LCS is that it allows the author to type in an answer that is general enough to match any number of additional answers.

### 3 Overview of LCS Acquisition

We use Levin's publicly available online index (Levin, 1993) as a starting point for building LCS-based verb entries.<sup>1</sup> While this index provides a unique and extensive catalog of verb classes, it does not define the underlying meaning components of each class. One of the main contributions of our work is that it provides a relation between Levin's classes and meaning components as defined in the LCS representation.

Table 2 shows three broad semantic categories and example verbs along with their associated LCS representations. We have hand-constructed a database containing 191 LCS templates, i.e., one for each verb class in (Levin, 1993). In addition, we have generated LCS templates for 26 additional classes that are *not* included in Levin's system. Several of these correspond to verbs that take sentential complements (e.g., *coerce*).

<sup>1</sup>We focus on building entries for verbs; however, we have approximately 30,000 non-verb entries per language.

Table 2: Sample Templates Stored in the LCS Database

Category	Verb	Class	Grid	LCS
Location	suspend	9.2	,ag_th,loc()	[CAUSE (X, [BE <sub>Loc</sub> (Y, [AT <sub>Loc</sub> (Y, Z)]), [BY (MANNER)])])]
	touch	47.8	_th_loc	[BE <sub>Loc</sub> (Y, [AT <sub>Loc</sub> (Y, Z)], [BY (MANNER)])]
Motion	abandon	51.2	_th_src	[GO <sub>Loc</sub> (Y, [(DIRECTION) <sub>Loc</sub> (Y, [AT <sub>Loc</sub> (Y, Z)])])]
	float	51.3.1	_th_src(),goal()	[GO <sub>Loc</sub> (Y, [BY (MANNER)])]
Placement	adorn	9.8	_ag_th,mod-poss(with)	[CAUSE (X, [GO <sub>Ident</sub> (Y, [TOWARD <sub>Ident</sub> (Y, [AT <sub>Ident</sub> (Y, [(STATE) <sub>Ident</sub> (([WITH] <sub>Poss</sub> (*HEAD*, Z)))])))])))]])]
	spill	9.5	,ag_th	[CAUSE (X, [GO <sub>Loc</sub> (Y)], [BY (MANNER)])]

A full entry in the database includes a semantic class number with a list of possible verbs, a thematic grid, and a LCS template:

(3) **Class 47.8:** adjoin, intersect, meet, touch, ...

**Thematic Grid:** \_th\_loc

**LCS Template:**

```
(be loc (thing 2)
(at loc (thing 2) (thing 11))
(!!-ingly 26))
```

The semantic class label 47.8 above is taken from Levin's 1993 book (*Verbs of Contiguous Location*), i.e., the class to which the verb *touch* has been assigned.<sup>2</sup> A verb, together with its semantic class uniquely identifies the word sense, or LCS template, to which the verb refers. The thematic grid (\_th\_loc) indicates that the verb has two obligatory arguments, a *theme* and a *location*.<sup>3</sup> The !! in the LCS Template acts as a wildcard; it will be filled by a lexeme (i.e., a root form of the verb). The resulting form is called a *constant*, i.e., the idiosyncratic part of the meaning that distinguishes among members of a verb class (in the spirit of (Grimshaw, 1993; Levin and Rappaport Hovav, *To appear*; Pinker, 1989; Talmy, 1985)).<sup>4</sup>

Three inputs are required for acquisition of verb entries: a semantic class, a thematic grid, and a lexeme, which we will henceforth abbreviate as "class/grid/lexeme." The output is a Lisp-like expression corresponding to the LCS representation. An example of input/output for our acquisition procedure is shown here:

(4) **Acquisition of LCS for:** *touch*

**Input:** 47.8: \_th\_loc; "touch"

<sup>2</sup>Verbs not occurring in Levin's book are also assigned to classes using techniques described in (Dorr and Jones, 1996; Dorr, *To appear*).

<sup>3</sup>An underscore (\_) designates an obligatory role and a comma (,) designates an optional role.

<sup>4</sup>The !! in the Lisp representation corresponds to the angle-bracketed constants in Table 2, e.g., !!-ingly corresponds to <MANNER>.

**Output:**

```
(be loc (* thing 2)
(at loc (thing 2) (* thing 11))
(touchingly 26))
```

Language-specific annotations such as the \*-marker in the LCS Output are added to the templates by processing the components of thematic grid specifications, as we will see in more detail next.

## 4 Language-Specific Annotations

In our on-going example (4), the thematic grid \_th\_loc indicates that the *theme* and the *location* are both obligatory (in English) and should be annotated as such in the instantiated LCS. This is achieved by inserting a \*-marker appropriately. Consider the structural divergence between the following English/Spanish equivalents:

(5) **Structural Divergence:**

```
E: John entered the house.
S: John entró a la casa.
'John entered into the house.'
```

The English sentence differs structurally from the Spanish in that the noun phrase *the house* corresponds to a prepositional phrase *a la casa*. This distinction is characterized by different positionings of the \*-marker in the lexical entries produced by LEXICALL:

(6) **Lexical Entries:**

```
enter: (go loc (* thing 2)
(toward loc (thing 2)
(in loc (thing 2) (* thing 6)))
(enteringly 26))
entrar: (go loc (* thing 2)
((* toward 5) loc (thing 2)
(in loc (thing 2) (thing 6)))
(enteringly 26))
```

The lexicon entries for *enter* and *entrar* both mean "X (= Thing 2) goes into location Y (= Thing 6)." Variable positions (designated by numbers, such as 2, 5 and 6) are used in place of the ultimate fillers

such as *john* and *house*. The structural divergence of (5) is accommodated as follows: the \*-marked leaf node, i.e., (*thing 6*) in the *enter* definition, is filled directly, whereas the \*-marked non-leaf node, i.e., ((*toward 5*) *loc ...*) in the *entrar* definition, is filled in through unification at the internal *toward* node.

## 5 Construction of Lexical Entries

Consider the construction of a lexical entry for the verb *adorn*. The LCS for this verb is in the class of *Fill Verbs* (9.8).<sup>5</sup>

```
(7) (cause (thing 1)
      (go ident (thing 2)
        (toward ident (thing 2)
          (at ident (thing 2) (!! -ed 9))))
      (with poss (*head*) (thing 16)))
```

This list structure recursively associates logical heads with their arguments and modifiers. The logical head is represented as a primitive/field combination, e.g., GO<sub>Ident</sub> is represented as (*go ident ...*). The arguments for CAUSE are (*thing 1*) and (*go ident ...*). The substructure GO itself has two arguments (*thing 2*) and (*toward ident ...*) and a modifier (*with poss ...*).<sup>6</sup> The *!!-ed* constant refers to a resulting state, e.g., adorned for the verb *adorn*. The LCS produced by our program for this verb is:

```
(8) (cause (thing 1)
      (go ident (thing 2)
        (toward ident (thing 2)
          (at ident (thing 2) (adorned 9))))
      (with poss (*head*) (thing 16)))
```

The variables in the representation map between LCS positions and their corresponding thematic roles. In the LCS framework, thematic roles provide semantic information about properties of the argument and modifier structures. In (7) and (8) above, the numbers 1, 2, 9, and 16 correspond to the roles agent (*ag*), theme (*th*), predicate (*pred*), and possessional modifier (*mod-poss*), respectively. These numbers enter into the construction of LCS entries: they correspond to argument positions in the LCS template (extracted using the class/grid/lexeme specification). Information is filled into the LCS template using these numbers, coupled with the thematic grid tag for the particular word being defined.

### 5.1 Fundamentals

LEXICALL locates the appropriate template in the LCS database using the class/grid pairing as an in-

<sup>5</sup>Some of the other 9.8 verbs are: *anoint*, *bandage*, *flood*, *frame*, *garland*, *stud*, *suffuse*, *surround*, *veil*.

<sup>6</sup>The *\*head\** symbol—used for modifiers—is a placeholder that points to the root (cause) of the overall lexical entry.

dex, and then determines the language-specific annotations to instantiate for that template. The default position of the \*-marker is the left-most occurrence of the LCS node corresponding to a particular thematic role. However, if a preposition occurs in the grid, the \*-marker may be placed differently. In such a case, a primitive representation (e.g., (*to loc (at loc)*)) is extracted from a set of predefined mappings. If this representation corresponds to a subcomponent of the LCS template, the program recognizes this as a match against the grid, and the \*-marker is placed in the template at the level where this match occurs (as in the entry for *entrar* given in (6) above).

If a preposition occurs in the grid but there is no matching primitive representation, the preposition is considered to be a collocation, and it is placed in a special slot—*:collocations*—which indicates that the LCS already covers the semantics of the verb and the preposition is an idiosyncratic variation (as in *learn about*, *know of*, etc.).

If a preposition is required but it is not specified (i.e., empty parentheses ()), then the \*-marker is positioned at the level dominating the node that corresponds to that role—which indicates that several different prepositions might apply (as in *put on*, *put under*, *put through*, etc.).

### 5.2 Examples

The input to LEXICALL is a class/grid/lexeme specification, where each piece of information is separated by a hash sign (#):

```
<class>#<grid>#<lexeme>#
<other semantic information>
```

For example, the input specification for the verb *replant* (a word not classified by Levin) is:

```
9.7#_ag_th,mod-poss(with)#replant#
!!-ed = planted (manner = again)
```

This input indicates that the class assigned to *replant* is 9.7 (Levin's Spray/Load verbs) and its grid has an obligatory agent (*ag*), theme (*th*), and an optional possessional modifier with preposition *with* (*mod-poss(with)*). The information following the final # is optional; this information was previously hand-added to the assigned thematic grids. In the current example, the *!!-ed* designates the form of the constant *planted* which, in this case, is a morphological variant of the lexeme *replant*.<sup>7</sup> Also, the

<sup>7</sup>The constant takes one of several forms, including: *!!-ingly* for a manner, *!!-er* for an instrument, and *!!-ed* for resulting states. If this information has not been hand-added to the class/grid/lexeme specification (as is the case with most of the verbs), a default morphological process produces the appropriate form from the lexeme.

manner *again* is specified as an additional semantic component.

For presentational purposes, the remainder of this section uses English examples. However, as we saw in Section 4, the representations used here carry over to other languages as well. In fact, we have used the same acquisition program, without modification, for building our Spanish and Arabic LCS-based lexicons, each of size comparable to our English LCS-based lexicon.

## I. Thematic Roles without Prepositions

(9) **Example:** The flower decorated the room.

**Input:** 9.8#\_mod-poss\_th#decorate#

**Template:**

```
(be ident (thing 2)
  (at ident (thing 2) (!! -ed 9))
  (with poss (*head*) (thing 16)))
```

Two thematic roles, *th* and *mod-poss*, are specified for the above sense of the English verb *decorate*. The thematic code numbers—2 and 16, respectively—are \*-marked and the constant *decorated* replaces the wildcard:

(10) **Output:**

```
(be ident (* thing 2)
  (at ident (thing 2) (decorated 9))
  (with poss (*head*) (* thing 16)))
```

## II. Thematic Roles with Unspecified Prepositions

(11) **Example:** We parked the car near the store.

We parked the car in the garage.

**Input:** 9.1#\_ag\_th\_goal()#park#

**Template:**

```
(cause (thing 1)
  (go loc (thing 2)
    (toward loc (thing 2)
      ([at] loc (thing 2) (thing 6))))
  (!! -ingly 26))
```

The input for this example indicates that the goal is headed by an unspecified preposition. The thematic roles *ag*, *th*, and *goal()* correspond to code numbers 1, 2, and 6, respectively. The variable positions for *ag* and *th* are \*-marked just as in the previous case, whereas *goal()* requires a different treatment. When a required preposition is left unspecified, the \*-marker is associated with a LCS node dominating a generic *[at]* position:

(12) **Output:**

```
(cause (* thing 1)
  (go loc (* thing 2)
    (!! toward 5) loc (thing 2)
    ([at] loc (thing 2) (thing 6))))
  (parkingly 26))
```

## III. Thematic roles with Specified Prepositions

(13) **Example:** We decorated the room with flowers.

**Input:** 9.8#\_ag\_th\_mod-poss(with)#decorate#

**Template:**

```
(cause (thing 1)
  (go ident (thing 2)
    (toward ident (thing 2)
      (at ident (thing 2) (!! -ed 9))))
  (with poss (*head*) (thing 16)))
```

Here, the *mod-poss* role requires the preposition *with* in the modifier position:

(14) **Output:**

```
(cause (* thing 1)
  (go ident (* thing 2)
    (toward ident (thing 2)
      (at ident (thing 2) (decorated 9))))
  (!! with 15) poss (*head*) (thing 16)))
```

In order to determine the position of the \*-marker for a thematic role with a required preposition, LEXICALL consults a set of predefined mappings between prepositions (or postpositions, in a language like Korean) and their corresponding primitive representations.<sup>8</sup> In the current case, the preposition *with* is mapped to the following primitive representation: (*with poss*). Since this matches a sub-component of the LCS template, the program recognizes this as a match against the grid, and the \*-marker is placed in the template at the level of *with*.

## 6 Limitations and Conclusions

We have described techniques for automatic construction of dictionaries for use in large-scale FLT. The dictionaries are based on a language-independent representation called *lexical conceptual structure* (LCS). Significant enhancements to LCS-based tutoring could be achieved by combining this representation with a mechanism for handling issues related to discourse and pragmatics. For example, although the LCS processor is capable of determining that the phrase *in the trash* partially matches the answer to *Where did John put the book?*, a pragmatic component would be required to determine that this answer is (perhaps) more appropriate than the full answer, *He put the book in the trash*. Representing conversational context and dynamic context updating (Traum et al., 1996; Haller, 1996; DiEugenio and Webber, 1996) would provide a framework for this type of response “relaxation.” Along

<sup>8</sup>We have defined approximately 100 such mappings per language. For example, the mapping produces the following primitive representations for the English word *to*: (*to loc (at loc)*), (*to poss (at poss)*), (*to temp (at temp)*), (*toward loc (at loc)*), (*toward poss (at poss)*). We have similar mappings defined in Arabic and Spanish. For example, the following primitive representations are produced for the Spanish word *a*: (*at loc*), (*to loc (at loc)*), (*to poss (at poss)*), (*toward loc (at loc)*).

these same lines, a pragmatic component could provide a mechanism for determining that certain fully matched responses (e.g., *John hurled the book into the trash*) are not as “realistic sounding” as partially matched alternatives.

Initially, LEXICALL was designed to support the development of LCS’s for English only; however, the same techniques can be used for multilingual acquisition. As the lexicon coverage for other languages expands, it is expected that our acquisition techniques will help further in the cross-linguistic investigation of the relationship between Levin’s verb classes and the basic meaning components in the LCS representation. In addition, it is expected that verbs in the same Levin class may have finer distinctions than what we have specified in the current LCS templates.

We view the importation of LCS’s from the English LCS database into Arabic and Spanish as a first approximation to the development of complete lexicons for these languages. The results have been hand-checked by native speakers using the class/grid/lexeme format (which is much easier to check than the fully expanded LCS’s). The lexical verification process took only two weeks by the native speakers. We estimate that it would take at least 6 months to build such a lexicon from scratch (by human recall and data entry alone), and in such a case, the potential for error would be at least twice as high.

One important benefit of using the Levin classification as the basis of our program is that, once the mapping between verb classes and LCS representations has been established, we can acquire the LCS representation for a new verb (i.e., one not in Levin) simply by associating it with one of the 191 classes. We see our approach as a first step toward compression of lexical entries in that it allows lexicons to be stored in terms of the more condensed class/grid/lexeme specifications; these can be expanded online, as needed, during sentence processing in the NLP application.

We conclude that, while human intervention is necessary for the acquisition of class/grid information, this intervention is virtually eliminated from the LCS construction process because of our provision of a mapping between semantic classes and primitive meaning components.

## Acknowledgements

I would like to thank Jungshin Park and Mine Ulku Sencan for their aid in the development of certain components of the LEXICALL program. In addition, comments from five anonymous reviewers greatly enhanced the presentation of this work. The author has been supported, in part, by Army Research Office contract DAAL03-91-C-0034 through Battelle Corporation, NSF NYI IRI-9357731 and Logos Corporation, NSF CNRS INT-9314583. Ad-

vanced Research Projects Agency and ONR contract N00014-92-J-1929, Alfred P. Sloan Research Fellow Award BR3336, Army Research Institute contract MDA-903-92-R-0035 and Microelectronics and Design, Inc., and the University of Maryland General Research Board.

## References

- Brent, Michael. 1993. Unsupervised Learning of Lexical Syntax. *Computational Linguistics*, 19:243–262.
- Carrier, Jill and Janet H. Randall. 1993. Lexical mapping. In Eric Reuland and Werner Abraham, editors, *Knowledge and Language II: Lexical and Conceptual Structure*. Kluwer, Dordrecht, pages 119–142.
- Church, Kenneth and P. Hanks. 1990. Word Association Norms, Mutual Information and Lexicography. *Computational Linguistics*, 16:22–29.
- Copestake, Ann, Ted Briscoe, P. Vossen, A. Ageno, I. Castellon, F. Ribas, G. Rigau, H. Rodriguez, and A. Samiotou. 1995. Acquisition of Lexical Translation Relations from MRDS. *Machine Translation*, 9:183–219.
- DiEugenio, Barbara and Bonnie Lynn Webber. 1996. Pragmatic Overloading in Natural Language Instructions. *International Journal of Expert Systems*, 9(1):53–84.
- Dorr, Bonnie J. To appear. Large-Scale Dictionary Construction for Foreign Language Tutoring and Interlingual Machine Translation. *Machine Translation*, 12(1).
- Dorr, Bonnie J., Joseph Garman, and Amy Weinberg. 1995. From Syntactic Encodings to Thematic Roles: Building Lexical Entries for Interlingual MT. *Machine Translation*, 9:71–100.
- Dorr, Bonnie J. and Douglas Jones. 1996. Role of Word Sense Disambiguation in Lexical Acquisition: Predicting Semantics from Syntactic Cues. In *Proceedings of the International Conference on Computational Linguistics*, pages 322–333, Copenhagen, Denmark.
- Dowty, David. 1991. The Effects of Aspectual Class on the Temporal Structure of Discourse: Semantics or Pragmatics? *Language*, 67:547–619.
- Fillmore, Charles. 1968. The Case for Case. In E. Bach and R. Harms, editors, *Universals in Linguistic Theory*. Holt, Rinehart, and Winston, pages 1–88.
- Foley, William A. and Robert D. Van Valin. 1984. *Functional Syntax and Universal Grammar*. Cambridge University Press, Cambridge.
- Grimshaw, Jane. 1990. *Argument Structure*. MIT Press, Cambridge, MA.

- Grimshaw, Jane. 1993. Semantic Structure and Semantic Content in Lexical Representation. unpublished ms., Rutgers University. New Brunswick, NJ.
- Gruber, Jeffrey S. 1965. *Studies in Lexical Relations*. Ph.D. thesis, MIT. Cambridge, MA.
- Hale, Ken and Samuel J. Keyser. 1993. On Argument Structure and Lexical Expression of Syntactic Relations. In Ken Hale and Samuel J. Keyser, editors, *The View from Building 20: Essays in Honor of Sylvain Bromberger*. MIT Press, Cambridge, MA.
- Haller, Susan. 1996. Planning Text About Plans Interactively. *International Journal of Expert Systems*, 9(1):85-112.
- Holland, Melissa. 1994. Intelligent Tutors for Foreign Languages: How Parsers and Lexical Semantics can Help Learners and Assess Learning. In *Proceedings of the Educational Testing Service Conference on Natural Language Processing Techniques and Technology in Assessment and Education*, Princeton, NJ: ETS.
- Jackendoff, Ray. 1983. *Semantics and Cognition*. MIT Press, Cambridge, MA.
- Jackendoff, Ray. 1990. *Semantic Structures*. MIT Press, Cambridge, MA.
- Jackendoff, Ray. 1996. The Proper Treatment of Measuring Out, Telicity, and Perhaps Even Quantification in English. *Natural Language and Linguistic Theory*, 14:305-354.
- Klavans, Judith L. and Evelynne Tzoukermann. 1995. Dictionaries and Corpora: Combining Corpus and Machine-Readable Dictionary Data for Building Bilingual Lexicons. *Machine Translation*, 10:185-218.
- Levin, Beth. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. Chicago, IL.
- Levin, Beth and Malka Rappaport Hovav. To appear. Building Verb Meanings. In M. Butt and W. Gauder, editors, *The Projection of Arguments: Lexical and Syntactic Constraints*. CSLI.
- Lonsdale, Deryle, Teruko Mitamura, and Eric Nyberg. 1995. Acquisition of Large Lexicons for Practical Knowledge-Based MT. *Machine Translation*, 9:251-283.
- Pesetsky, David. 1982. *Paths and Categories*. Ph.D. thesis, MIT, Cambridge, MA.
- Pinker, Steven. 1989. *Learnability and Cognition: The Acquisition of Argument Structure*. MIT Press, Cambridge, MA.
- Sams, Michelle. 1993. An Intelligent Foreign Language Tutor Incorporating Natural Language Processing. In *Proceedings of Conference on Intelligent Computer-Aided Training and Virtual Environment Technology*, NASA: Houston, TX.
- Talmy, Leonard. 1985. Lexicalization Patterns: Semantic Structure in Lexical Forms. In T. Shopen, editor, *Language Typology and Syntactic Description 3: Grammatical Categories and the Lexicon*. University Press, Cambridge, England, pages 57-149.
- Traum, David R., Lenhart K. Schubert, Nathaniel G. Martin, Chung Hee Hwang, Peter Heeman, George Ferguson, James Allen, Massimo Poesio, and Marc Light. 1996. Knowledge Representation in the TRAINS-93 Conversation System. *International Journal of Expert Systems*, 9(1):173-223.
- Weinberg, Amy, Joseph Garman, Jeffery Martin, and Paola Merlo. 1995. Principle-Based Parser for Foreign Language Training in German and Arabic. In Melissa Holland, Jonathan Kaplan, and Michelle Sams, editors, *Intelligent Language Tutors: Theory Shaping Technology*. Lawrence Erlbaum Associates, Hillsdale, NJ.
- Wu, D. and X. Xia. 1995. Large-Scale Automatic Extraction of an English-Chinese Translation Lexicon. *Machine Translation*, 9:285-313.