DEVELOPMENT OF INTERLINGUAL LEXICAL CONCEPTUAL STRUCTURES WITH SYNTACTIC MARKERS FOR MACHINE TRANSLATION

UMIACS TR 95-16, CS TR 3412

by

Bonnie Dorr, Jye-hoon Lee, Clare Voss, and Sungki Suh
Institute for Advanced Computer Studies
and Department of Computer Science
University of Maryland
A.V. Williams Building
College Park, MD 20742

for

U.S. Army Research Office
Center for Command, Control, and Communications Systems (C3)
(Mr. George Yaeger)

February, 1995

Contract No. DAAL03-91-C-0034
TCN Number 93005
Scientific Services Program

Abstract

This document reports on research conducted at the University of Maryland for the Korean/English Machine Translation (MT) project. Our primary objective was to develop an interlingual representation based on lexical conceptual structure (LCS) and to examine the relation between this representation and a set of linguistically motivated semantic classes. We view the work of the past year as a critical step toward achieving our goal of building a generator: the classification of LCS's into a semantic hierarchy provides a systematic mapping between semantic knowledge about verbs and their surface syntactic structures.

We have focused on several areas in support of our objectives: (1) investigation of morphological structure including distinctions between Korean and English; (2) porting a fast, message-passing parser to Korean (and to the IBM PC); (3) study of free word order and development of the associated processing algorithm; (4) investigation of the aspectual dimension as it impacts morphology, syntax, and lexical semantics; (5) investigation of the relation between semantic classes and syntactic structure; (6) development of theta-role and lexical-semantic templates through lexical acquisition techniques; (7) definition a mapping between KR concepts and interlingual representations; (8) formalization of the lexical conceptual structure

Foreword

This document reports on research conducted at the University of Maryland for the Korean/English Machine Translation (MT) project. This effort has involved Dr. Bonnie Dorr, professor in computational linguistics, and three graduate students in computer science. Biographical sketches are given below. In addition, the Maryland team is in close collaboration with Martha Palmer at the University of Pennsylvania and Patrick Saint-Dizier at IRIT in Toulouse, France.

Bonnie Dorr received her bachelor's degree, summa cum laude, in computer science from Boston University, Boston, MA, in 1984, and her S.M. and Ph.D. in computer science from the Massachusetts Institute of Technology, Cambridge, MA, 1990. Between September 1990 and June 1992, she was a research associate with the Institute for Advanced Computer Studies at the University of Maryland, College Park; since then she has been an assistant professor in the Computer Science Department. She has over a decade of experience with natural language processing systems, has been a part of the natural language processing group at the University of Maryland since 1990, and has undertaken extensive practical industry consulting as well as work with multilingual translation development and evaluation at MITRE. Dr. Dorr has recently received the Alfred P. Sloan Research Fellowship and a NSF Young Investigator Award for her work on Large-Scale Interlingual Machine Translation. She is the author of several published papers on natural language processing as well as a book on the lexicon in machine translation. She is a member of ACL, ACM, AMTA, AAAI, Phi Beta Kappa, and Pi Mu Epsilon, and Sigma Xi.

Jye-hoon Lee received his bachelor's degree in mechanical engineering at Seoul National University, Seoul, Korea in 1989, studied Computer Science at Iowa State University between 1989 and 1991 and then became a doctoral student in the computer science graduate program at the University of Maryland. Since January 1993, Mr. Lee has been involved in the use of corpora for the acquisition of lexical-semantic features associated with verb meaning in the context of machine translation. Over the last year, he has also written a transliteration program from Yale romanization to Korean (and vice versa) and has analyzed problems that arise in translating between Korean and English. Mr. Lee expects to graduate in 1996.

Sungki Suh received his bachelor's degree in English language and literature at Seoul National University, Seoul, Korea in 1986 and his M.A. in linguistics at State University of New York at Buffalo, New York. He received his Ph.D. in Linguistics, graduating 3 months prior to the completion of the project in 1994. He is currently a professor in the Language Research Institute at Seoul National University. Dr. Suh specialized in the area of computational linguistics and is one of the few linguists in the world whose expertise is in Korean syntax. He has recently published papers in prestigious linguistics conferences and has extensive research experience in the linguistics department at Maryland.

Clare Voss received her bachelor's degree in linguistics at the University of Michigan, Ann Arbor, MI, her M.A. in psychology at the University of Pennsylvania, Philadelphia, PA, and is currently a doctoral student in the computer science department at the University of Maryland. Ms. Voss has been a researcher in computational linguistics for over seven years and has specialized in the cross-linguistic study of diverse languages such as French, German, Portuguese, and Estonian. She is currently working on a lexicon-based machine translation system called LEXITRAN under Dr. Dorr, which has a comprehensive coverage of spatial and aspectual relations and uses the principles-and-parameters framework assumed for the current project. She is also designing a MT developer's tool called ILustrate which allows computational linguists to test different version of the IL representation. Ms. Voss expects to receive her Ph.D. in fall of 1996.

Martha Palmer is an adjunct professor in the Institute for Research in Cognitive Science at the University of Pennsylvania. Beginning as a Sloan Fellow at the University of Pennsylvania in 1981, and continuing as a Senior Research Scientist at Unisys in the 80's, Dr. Palmer has investigated the formalizations of underlying conceptual representations of verbs in several different types of subdomains, including physics word problems and a natural language interface to a graphics system. During a 3-year stay in Singapore, Dr. Palmer extended her methods of verb representation to French and Chinese, and investigated their utility for resolving cross-linguistic divergences. More recently, she has been collaborating with Dr. Dorr and Dr. Saint-Dizier on conceptual representations of English and French verbs that are necessary for accurate lexical selection in machine translation. She has also investigated the application of lexicalized grammars to the syntactic and semantic structure of Korean sentences.

Patrick Saint-Dizier has directed work in natural language at the Universite Paul Sabatier in Toulouse, France since 1986, laying the foundations for addressing problems in understanding and generation of language. After collecting real world data and discovering that certain linguistic constructs and lexical items recur regularly in different languages, Dr. Saint-Dizier's group has pursued a formal characterization of the linguistic theory of the lexicon and has developed constraint logic programming based techniques for translating natural language based on this formalism.

Acknowledgments

This work was supported by the Center for Command, Control, and Communications Systems (C3) (Mr. George Yaeger) under the auspices of the U.S. Army Research Office Scientific Services Program administered by Battelle (Delivery Order 584, Contract No. DAAL03-91-C-0034).

Contents

1 Introduction

Over the last year, the computational linguistics group at Maryland has been involved in the problem of interlingual machine translation of Korean and English. We have adopted a pivot form called Lexical Conceptual Structure (LCS), was developed on the basis of work by (Jackendoff, 1983), (Jackendoff, 1990). Our primary objective was to develop an interlingual representation based on lexical conceptual structure (LCS) and to examine the relation between this representation and a set of linguistically motivated semantic classes. We view the work of the past year as a critical step toward achieving our goal of building a generator: the classification of LCS's into a semantic hierarchy provides a systematic mapping between semantic knowledge about verbs and their surface syntactic structures.

We have focused on several areas in support of our objectives: (1) investigation of morphological structure including distinctions between Korean and English; (2) porting a fast, message-passing parser to Korean (and to the IBM PC); (3) study of free word order and development of the associated processing algorithm; (4) investigation of the aspectual dimension as it impacts morphology, syntax, and lexical semantics; (5) investigation of the relation between semantic classes and syntactic structure; (6) development of theta-role and lexical-semantic templates through lexical acquisition techniques; (7) definition a mapping between KR concepts and interlingual representations; (8) formalization of the lexical conceptual structure Each of these areas is addressed, in turn, in the following sections.

2 Transliteration and Morphology

We continued development on the Hangul Converter and also made some headway on developing rules that will serve as input to the Kimmo system for a morphological analyzer/synthesizer.

2.1 Hangul Converter

At the beginning of 1994, we finalized the program that allows us to convert between Hangul and romanized forms. Appendix A provides usage notes for this program. This year, we have added the ability to use Chinese characters. There exist more than 100,000 Chinese characters. Because there is no alphabet system in Chinese, this facility requires a 17-bit coding system to distinguish among all possibilities. ($2^17 = 131072$)

KSC-5601, which is a complete Hangul code, defines most commonly used 3000 Chinese characters. However, the KSC-5601 code was defined in Korea; thus, no other Chinese characters are allowed other than 3000 Chinese characters defined in KSC-5601.

A difficulty that we faced when we installed Chinese characters is that there is a many-many mapping between Chinese characters and Korean characters. Chinese characters have potentially more than one Korean sound, and Korean characters, which collectively have 2350 different sounds, can be mapped to many different Chinese characters. Thus we were forced to build, by hand, a very large Chinese character table. Our approach was to distinguish among several Chinese characters of the same sound using a signal for each case (e.g., Ka1, Ka2, ...).

We added an external romanization mapping table that allows the user to define mapping functions between a 2 byte character and a given romanization scheme. We added 3000 Chinese characters in the 5601 standard.

2.2 Irregular Verbs

We developed the following 6 rules of morphology for irregular verbs using (Ihm et al., 1988).

1. `n`

```
l --> 0 / __ +  s
                p
```

2.
   ```
   p --> o / o __ + V(owel)
   p --> wu / __ + V
   Exceptions: nelp-ta, cop-ta, cip-ta, cap-ta, pwutcap-ta,
   ip-ta, ep-ta, ppop-ta, ssip-ta
   ```

3.
   ```
   t --> l / __ + V
   Exceptions: mwut-ta, mit-ta, et-ta, ssot-ta, pat-ta, tat-ta
   ```

4.
   ```
   u --> l / l __ +  a
                     e
   ```

5.
   ```
                               n
   h --> 0 / #(C)V(C)CV __ +  l
                               m
                               s
   ```

6.
   ```
   s --> 0 / __ + V
   Exceptions: ppayas-ta, wus-ta, pes-ta, ssis-ta
   ```

2.3 Nominal Suffix

We undertook an analysis of the nominal suffix from (Ihm et al., 1988), which uncovered the following information:

I. Case Marker

  (1) Nominative case marker: ka/ i, kkeyse(-honorific)
  `ka --> i / C(onsonant) + ___`
  (2) Accusative case marker: lul/ ul
  `lul --> ul / C + ___`
  (3) Genitive case marker: uy

II. Postposition

  (1) Instrumental: lo/ ulo
  `lo --> ulo / C + ___ (Exception: ulo --> lo / l + ___ )`
  (2) Reason or Source: lo/ ulo, ey
  (3) Status: lo/ ulo
  (4) Resultative: lo/ ulo
  (5) Locative
    (a) (default): ey
    (b) [+Animate] (=Dative): ey-key, hanthey, kkey (-honorific)
    (c) Source: ey-se, ey-key-se, hanthey-se, pwuthe (-beginning point)
    *(d) Direction: lo/ ulo
    (e) Ending point: kkaci
    (f) Eventive: eyse
  *(6) Temporal

    - (a) Eventive: ey
    - (b) Beginning point: eyse, pwuthe
    - (c) Ending point: kkaci
  - (7) Measure: ey
  - (8) Commitative: wa/ kwa, lang/ ilang, hako
    `wa --> kwa / C + ___`
    `lang --> ilang / C + ___`

III. Delimiter

  - (1) Topic: nun/ un
    `nun --> un / C + ___`
    cf: Topic marking of the subject is required when the sentence is classified as a depictive statement.
  - (2) Only: man
  - (3) Too: to
  - (4) Even: cocha, mace
  - (5) Each: mata
  - (6) Unselective or Emphasis: na/ ina
    `na --> ina / C + ___`
  - (7) Amount: (Only in interrogative sentences): na/ ina

** When a delimiter is attached to the subject/ object NP, nominative/ accusative case marker on the NP must be deleted. (Case III.(2) above is an exception.)

```
• John-un /*John-i-un    Mary-to/ *Mary-ul-to    coahanta
      -Top     -Nom-Top     -too      -Acc-too      like
  'John likes Mary, too.'
  ('John, even Mary likes him.')
```

** Delimiters can be attached to adverbs/ verbs as well as nouns.

IV. Others

  - (1) Comparative: pota, kathi, chelem
  - (2) Coordinate conjunction
    - (a) And: wa/ kwa, lang/ ilang, hako
    - (b) Or: na/ ina

2.4 Verbal Endings

We undertook an analysis of Verbal Endings from (Ihm et al., 1988). In the description below, unless specified otherwise, the initial vowel of the verbal ending (= 'u' or 'e') is deleted when the verbal stem ends with a vowel. (Verbs whose stem ends with '-o'/'-wu' sound are exceptions to this generalization.) Also, the initial vowel 'e' of verbal ending changes to 'a' when the last syllable of verbal stem includes '-a' or '-o' sound.

I. Terminative Endings

*Terminative endings represent the mood type of sentences. They are classified based on the speech level. Speech level is mainly determined by the hearer's age, social status etc. relative to the speaker's.

- (1) Declarative
  - (a) [super high]: 'upnita'
  - (b) [high]: 'eyo', 'ciyo'
  - (c) [mid-low]: 'so', 'ne'
  - (d) [low]: 'ta', 'e'
- (2) Interrogative
  - (a) [super high]: 'upnikka'
  - (b) [high]: 'eyo', 'nayo'
  - (c) [mid-low]: 'swu', 'na'
  - (d) [low]: 'ni', 'nya'
- (3) Imperative
  - (a) [super high]: 'useyyo'
  - (b) [high]: 'eyo'
  - (c) [mid-low]: 'key'
  - (d) [low]: 'ela', 'e'
- (4) Propositive
  - (a) [super high]: 'siciyo'
  - (b) [high]: 'upsita', 'ciyo'
  - (c) [mid-low]: 'use'
  - (d) [low]: 'ca'

II. Adnominal Endings (-involving either a relative clause or a complex NP clause)

- (1) Present tense
  - (a) 'un': for adjectival verbs
  - (b) 'nun': otherwise

    -
      ```
      kem-un   koyangi
      black-Adnm   cat
      'A cat which is black'/'a black cat'
      ```

    -
      ```
      chayk-ul   ilk-nun  salam
       book-Acc read-Adnm  man
      'A man who is reading a book'
      ```
- (2) Future tense: 'ul'
- (3) Past tense
  - (a) (ess)'ten': implying reminiscence
  - (b) 'un': used only with non-adjectival verbs

III. Adverbial Endings

(1) reason or cause: ‘se’ ‘nikka’ `--->`for, as
(2) weak contrast: ‘nuntey’ `--->`while
(3) conditional: ‘umyen’ ‘ketun’ `--->`if, when
(4) purpose: ‘lyeko’ ‘le’ `--->`in order to (do), for (do)ing
(5) prerequisite:‘(e/a)ya’ ‘(e/a)yaman’ `--->`only when, only if
(6) goal: ‘tolok’ `--->`so that (one) may/can (do)
(7) concurrence: ‘umyense’ ‘umye’
(8) contrast: ‘ciman’ `--->`although (cf: coordinate conjunction ‘ciman’)
(9) separate action: ‘ta(ka)’
(10) greater degree:‘ulswulok’ `--->`the more (....the more)
(11) immediate sequence: ‘ca’ ‘camaca’ `--->` as soon as

IV. Nominal Endings -’um’ and ‘ki’

(1) [+tense]: ‘um’
(2) [-tense]: ‘ki’

*‘um’ must be accompanied by a case marker.

*‘um’ generally occurs with factive predicate, and ‘ki’ occurs with nonfactive predicate.

V. Coordinate Conjunction

(1) and: ‘ko’, ‘se’
*‘se’ is used when the first conjunct precedes the second one in time sequence or when the first conjunct is subordinate to the second one. (cf:‘kose’)
(2) or: ‘kena’
(3) but: ‘ciman’, ‘una’

VI. (Quotative) Complementizer: ‘ko’
*‘ko’ must be preceded by terminative endings

2.5 Korean Auxiliary Verbs

Our analysis of Korean auxiliary verbs from (Ihm et al., 1988) revealed that such verbs are classified primarily into two groups: One corresponds to an aspectual specification, and the other corresponds to the representation of a state which is different from the present.

I. Aspectual Specification
*V = main verb stem

(1a) V-e ‘peli-ta’: completion
(1b) V-e ‘nay-ta’: accomplishment
(1c) V-ko ‘mal-ta’: perfective(?)(-something is done at last)
(2a) V-e ‘noh-ta’: completion + duration
(2b) V-e ‘twu-ta’: duration
(2c) V-e ‘kaci-ko’: duration(?) (-must be used in the form of ‘conjunction’)

(3a) V-kon ‘ha-ta’: habitual

(3b) V-e ‘tay-ta’: repetition(?)

(4) V-ko ‘iss-ta’: progressive

II. The State differing from the present

(1) V-ko ‘siph-ta’: hope (-want/hope to V)

(2) V-na ‘siph-ta’: (speaker’s) guess

(3) V-nunka ‘ha-ta’: (speaker’s) guess
V-na ‘ha-ta’

(4) V-un ‘tus-siph-ta’: (speaker’s) expect/ guess
V-ul ‘tus-siph-ta’

(5) V-un ‘tus-ha-ta’: (speaker’s) expect/ guess
V-ul ‘tus-ha-ta’

(6) V-eya ‘ha-ta’: obligation (-have to V)

(7) V-un ‘cheyha-ta’: pretense (-pretend to V)

(8) V-ul ‘ppenha-ta’: almost (-almost did something, but no success/ completion) `--->` past tense required

(9) V-ulye(ko) ‘ha-ta’: is about to V

(10) V-koca ‘ha-ta’: volition

(11) V-ulkka ‘ha-ta’: plan(-not decisive)

(12) V-ul ‘manha-ta’: is worthwhile to V

(13) V-na ‘po-ta’: (speaker’s) guess `--->`tense marker is not allowed

III. Others

(1) V-e ‘cwu-ta’: of benefit (-did something for others)

(2) V-e ‘po-ta’: trial

(3) V-ci ‘anh-ta’: negation (-does not V)

3 Message-Passing Parser and LCS Composition

Another area that we have investigated over the past year is that of efficient cross-linguistic parsing. The model we have adopted is a Message-Passing approach, based on Government-Binding (GB) Theory (Chomsky, 1981), (Haegeman, 1991), (van Riemsdijk and Williams, 1986). One of the drawbacks to alternative GB-based parsing approaches is that they generally adopt a filter-based paradigm. These approaches typically generate all possible candidate structures of the sentence that satisfy $\overline{\mathrm{X}}$ theory, and then subsequently apply filters in order to eliminate those structures that violate GB principles. (See, for example, (Abney, 1989), (Correa, 1991), (Dorr, 1991), (Fong, 1991), (Frank, 1990).) The current approach provides an alternative to filter-based designs which avoids these difficulties by applying principles to *descriptions* of structures without actually building the structures themselves. Our approach is similar to that of (Lin, 1993), (Lin and Goebel, 1993) in that structure building is deferred until the descriptions satisfy all principles; however, the current approach differs in that it provides a parameterization mechanism along the lines of (Dorr, 1993a) that allows the system to be ported to languages other than English. We focus particularly on the problem of processing head-final languages such as Korean.

We are currently incorporating the parser into our machine translation (MT) system, which we have now named PRINCITRAN.[1] In general, parsers of existing principle-based interlingual MT systems are exceedingly inefficient since they tend to adopt the filter-based paradigm. We combine the benefits of the message-passing paradigm with the benefits of the parameterized approach to build a more efficient, but easily extensible system, that will ultimately be used for MT. The algorithm has been implemented in C++ and successfully tested on well-known, translationally divergent sentences.

It is important to point out that the efficiency of the system is not simply a side effect of using an efficient programming language (i.e., C++), but that the algorithm is inherently efficient, independent of the programming language used for the implementation. The key to the efficiency of the algorithm is that it is based on a system that is already provably fast (Lin and Goebel, 1993). A formal, worst-case analysis reveals that the original CFG parsing algorithm is $O(|G|n^3)$ (Lin and Goebel, 1993), where n is the length of the input sentence and $|G|$ is the size of the grammar (the number of occurrences of non-terminal symbols in the set of grammar rules). This is an improvement of a factor of $|G|$ over standard CFG algorithms such as the Earley parser (Barton et al., 1987). This complexity measure is based on a serial simulation of parallel distributed message passing.

The extended version of the system differs only in that it includes attribute values, constraints, and movements, thus allowing a wider range of phenomena to be handled while still retaining the efficiency of the original algorithm. The complexity of this version has not yet been formally determined; however, in this article we will provide an intuitive sense of the inherent speed of this version by describing a set of experiments that show that the average time does not grow exponentially to the sentence length.[2]

The next section presents a general framework for parsing by message passing. The following section describes our implementation of GB principles as attribute-value constraints in the message-passing framework. We then present the parameterization framework, demonstrating the feasibility of handling cross-linguistic variation within the message-passing framework. A technique for automatic precompilation of parameter settings is then described. In the section on Time Test Comparisons, we compare the efficiency of the parser to that of the original CFG algorithm (Lin and Goebel, 1993) as well as Tomita's algorithm (Tomita, 1986) on a test suite of representative sentences. We conclude with a discussion about the implications of this approach and its applicability to the problem of multi-language MT; preliminary results on translationally divergent sentences in Korean and English are presented.

3.1 Message Passing Paradigm

There has been a great deal of interest in exploring new paradigms of parsing, especially non-traditional parallel architectures for natural language processing (Cottrell, 1989), (Waltz and Pollack, 1985), (Small, 1981), (Selman and Hirst, 1985), (Abney and Cole, 1985), (Jones, 1987), (Kempen and Vosse, 1989) (among many others). Recent work (Stevenson, 1994) provides a survey of symbolic, non-symbolic, and hybrid approaches. Stevenson's model comes the closest in design to the current principle-based message-passing model in that it uses distributed message passing as the basic underlying mechanism and it encodes GB principles directly (i.e., there are precise correspondences between functional components and linguistic principles). However, the fundamental goals of the two approaches are different: Stevenson's objective concerns the modeling of human processing behavior and producing a single parse at the end. Her system incorporates

---

[1] The name PRINCITRAN is derived from the names of two systems, UNITRAN (Dorr, 1993b) and PRINCIPAR (Lin, 1993).

[2] Presumably, runtime differences that are due solely to differences in efficiency of programming language will result in a curve that is similar to others, but at a different scale. We will show that, in fact, this is the result we obtained.

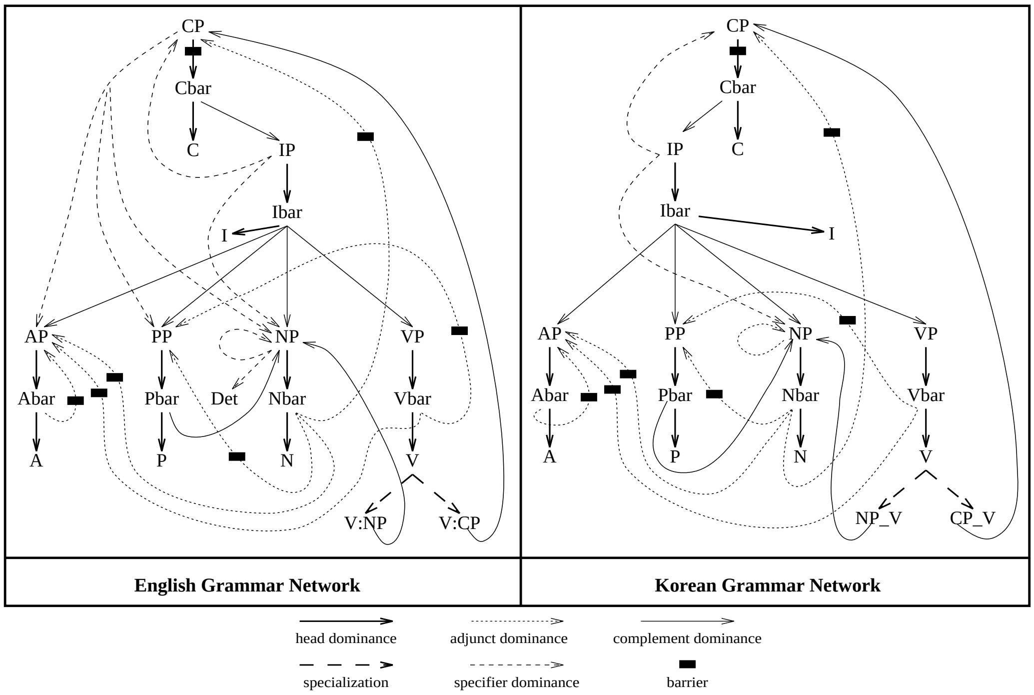


Figure 1: Network Representation of English and Korean Grammar

a number of psycholinguistic-based processing mechanisms for handling ambiguity and making attachment decisions.[3] Our model, on the other hand, is more concerned with efficiency issues, broad scale coverage, and cross-linguistic applicability; we produce all possible parse alternatives wherever disambiguation requires extra-sentential information.

We view Stevenson's approach as a special case of our own in that it would be possible to program our system so that it provides identical output for the same phenomena. The benefit of Stevenson's approach is that the linguistic constraints fall out from the machinery whereas our own design requires the constraints to be explicitly encoded in the processing data structures. On the other hand, a critical deficiency of Stevenson's design is that it is incapable of handling head-final languages unless the machinery undergoes a massive modification.

Our approach provides a language-independent processing mechanism that accommodates structurally different languages (e.g., head-initial vs. head-final) with equally efficient run times and it does not require major modifications to the underlying processing mechanism as each language is added. The approach extends the GB-based message-passing design proposed by (Lin, 1993), (Lin and Goebel, 1993). The grammar for each language is encoded as a network of nodes that represent grammatical categories (e.g., NP, Nbar, N) or subcategories, such as V:NP (i.e., a transitive verb that takes an NP as complement). Figure 1 depicts portions of the grammar networks used for English and Korean.

There are two types of links in the network: **subsumption links** (e.g., V to V:NP) and **dominance links** (e.g., Nbar to N). A dominance link from α to β is associated with an integer id that determines the linear order between β and other categories immediately dominated by α, and a binary attribute to specify whether β is optional or obligatory.[4]

---

[3] Stevenson's system currently handles structural ambiguity, but not lexical ambiguity; her research has focused almost entirely on the question of attachment decisions rather than on issues concerning selection of lexical items.

[4] For the purpose of readability, we have omitted integer id's in the graphical representation of the grammar network. Linear ordering is indicated by the starting points of links. For example, C precedes IP in the English network of figure 1.

Input sentences are parsed by passing messages in the grammar network. The nodes in the network are computing agents that communicate with each other by sending messages in the reverse direction of the links. Each node locally stores a set of items. An item is a triplet that represents a $\overline{\mathrm{X}}$ structure α: <surface-string, attribute-values, source-messages>, where *surface-string* is an integer interval [i,j] denoting the i'th to j'th word in the input sentence; *attribute-values* specifies syntactic features of the root node (β); and *source-messages* is a set of messages that represent immediate constituents of β and from which this item is combined. Each node has a completion predicate that determines whether an item at the node is "complete," in which case the item is sent as a message to other nodes.

When a node receives an item, it attempts to form new items by combining it with items from other nodes. Two items <[i_1,j_1], A_1, S_1> and <[i_2,j_2], A_2, S_2> can be combined if:

1. their surface strings are adjacent to each other: $i_2 = j_1+1$.
2. their attribute values A_1 and A_2 are unifiable.
3. the source messages come via different links: links(S_1) $\cap$ links(S_2) $= \emptyset$, where links(S) is a function that, given a set of messages, returns the set of links via which the messages arrived.

The result of the combination is a new item:

<[i_1,j_2], unify(A_1, A_2), $S_1 \cup S_2$>.

The new item represents a larger $\overline{\mathrm{X}}$ structure resulting from the combination of the two smaller ones. If the new item satisfies local constraints (to be described in the next section), it is considered valid and saved in the local memory. Otherwise, it is discarded. A valid item satisfying the completion predicate of the node is sent further as a message to other nodes.

More specifically, the parsing algorithm consists of the following steps:

Step 1: Lexical Look-up Retrieve the lexical entries for all the words in the sentence and create a lexical item for each word sense. A lexical item is a triple: <[i,j], av_{self}, av_{comp}>, where [i,j] is an interval denoting the position of the word in the sentence; av_{self} is the attribute values of the word sense; and av_{comp} is the attribute values of the complements of the word sense.

Step 2: Message Passing For each lexical item <[i,j], av_{self}, av_{comp}>, create an initial message <[i,j], av_{self}, $\emptyset$> and send this message to the grammar network node that represents the category or subcategory of the word sense. When the node receives the initial message, it may forward the message to other nodes or it may combine the message with other messages and send the resulting combination to other nodes. This initiates a message passing process which stops when there are no more messages to be passed around. At that point, the initial message for the next lexical item is fed into the network.

Step 3: Build a Shared Parse Forest When all lexical items have been processed, build a shared parse forest for the input sentence by tracing the origins of the messages at the highest node (CP or IP), whose `surface-string` component is the whole sentence. The links in the parse forest are derived from the grammar network links that are traversed during the tracing process. The structure of the parse forest is similar to (Tomita, 1986) and (Billot and Lang, 1989), but extended to include attribute values.

The parse trees representing the input sentence are retrieved from the parse forest one by one. The next section explains how constraints attached to the nodes and links in the network ensure that the parse trees satisfy all GB principles.

3.2 Implementation of Principles

GB principles are implemented as **local constraints** attached to nodes and **percolation constraints** attached to links. All items at a node must satisfy the node's local constraint. A message can be sent across a link only if it satisfies the link's percolation constraint. The idea of constraint application through feature passing among nodes is analogous to techniques applied in the TINA spoken language system (Seneff, 1992) except that, in our design, the grammar network is a *static* data structure; it is not *dynamically* modified during the parsing process. Thus, we achieve a reduction space requirements. Moreover, our design achieves a reduction in time requirements because we do not retrieve a structure until the resulting parse descriptions satisfy all the network constraints.

We will discuss three examples to illustrate the general idea of how GB principles are interpreted as local and percolation constraints. See (Lin, 1993) for more details.

3.2.1 $\overline{\mathrm{X}}$ Theory

The central idea behind $\overline{\mathrm{X}}$ theory is that a phrasal constituent has a layered structure. Every phrasal constituent is considered to have a head ($\mathrm{X}^0 \equiv \mathrm{X}$), which determines the properties of the phrase containing it. A phrase potentially contains a complement, resulting in a one-bar level ($\overline{\mathrm{X}} \equiv \mathrm{Xbar}$) projection; it may also contain a specifier (or modifier), resulting in a double-bar level ($\overline{\overline{\mathrm{X}}} \equiv \mathrm{XP}$) projection. The phrasal representation assumed in the current framework is the following:

(1) $[_{\mathrm{XP}}$ Specifier $[_{\mathrm{Xbar}}$ Complement X$]]$

We implement the relative positioning of Specifier, Complement, and Head constituents by means of dominance links as shown in each of the networks of Figure 1. In addition, adjuncts are associated with the Xbar level by means of an adjunct-dominance link in the grammar network.

The structure in (1) represents the relative order observed in Korean. We will see shortly that the linear order between the head X and the Complement is determined by a parameter of variation.

3.2.2 Trace Theory

A trace represents a position from which some element has been extracted.[5] The main constraint of Trace Theory is the Subjacency Condition, which prohibits movement across "too many" barriers. (The notion of "too many" is specified on a per-language basis as we will see shortly.)

An attribute named `barrier` is used to implement this principle. A message containing the attribute value `-barrier` is used to represent an $\overline{\mathrm{X}}$ structure containing a position out of which a wh-constituent has moved, but without yet crossing a barrier. The value `+barrier` means that the movement has already crossed one barrier. Certain dominance links in the network are designated as barrier links (indicated in Figure 1 by solid rectangles). The Subjacency condition is implemented by the percolation constraints attached to the barrier links, which blocks any message with `+barrier` and changes `-barrier` to `+barrier` (i.e., it allows the message to pass through).

3.2.3 Case Theory

Case theory requires that every NP be assigned abstract case. The Case Filter (Chomsky, 1981) rules out sentences containing an NP with no case. Case is assigned structurally to a syntactic position governed by a case assigner. Roughly, a preposition assigns Oblique Case to a prepositional object NP; a transitive verb assigns Accusative Case to a direct object NP; and tensed Infl(ection) assigns Nominative Case to a subject NP.

---

[5] A trace is represented as t_i, where i is a unique index referring to an antecedent.

Feature	**Significance**
`+ca`	the head is a case assigner
`-ca`	the head is not a case assigner
`+govern`	the head is a governor
`-govern`	the head is not a governor
`-cm`	an NP m-commanded by the head needs case marking

Figure 2: Case Theory Attribute Values

Node	**Local Constraint**
P	assign `+ca` to every item
V	assign `+ca` to items with `-passive`
I	assign `+ca` to items with `tense` attribute

Figure 3: Constraints on Case Assignment

The implementation of case theory in our system is based on the following attribute values: `ca, govern, cm` (see Figure 2). The attribute values `+ca` and `+govern` are assigned by local constraints to items representing phrases whose heads are case assigners (e.g., tensed I) and governors (e.g., V), respectively (see Figure 3). The Case Filter is then applied by checking the co-occurrence of the attributes `ca, govern,` and `cm`.

3.3 Implementation of Parameters

While the principles described in the previous section are intended to be language independent, the structure of each grammar network in figure 1 is too language-specific to be applicable to languages other than the one for which it is designated. The most obvious language-specific feature is the ordering of head links with respect to complement links; in the graphical representation link ordering of this type is indicated by the starting points of links, e.g., C precedes IP under Cbar since the link leading to C is to the left of the link leading to IP. In the English network, all phrasal heads precede their complements. However, in head-final languages such as Korean, the reverse order is required. In order to capture this distinction, we incorporate the parameterization approach of (Dorr, 1993a) into the message-passing framework so that grammar networks can be automatically generated on a per-language basis.

The reason the message-passing paradigm is so well-suited to a parameterized model of language parsing is that, unlike head-driven models of parsing, the main message-passing operation is capable of combining two nodes (in any order) in the grammar network. The result is that a head-final language such as Korean is as efficiently parsed as a head-initial language such as English. What is most interesting about this approach is that model is consistent with experimental results (see, for example, (Suh, 1993)) which suggest that constituent structure is computed prior to the appearance of the head in Korean.

The remainder of this section describes our approach to parameterization of each subtheory of grammar described in the last section; we conclude with a summary of the syntactic parameter settings for English and Korean.

3.3.1 $\overline{\text{X}}$ Theory

$\overline{\text{X}}$ theory assumes that a constituent order parameter is used for specifying phrasal ordering on a per-language basis:

(2) **Constituent Order:** The relative order between the head and its complement can vary, depending on whether the language in question is (i) head-initial or (ii) head-final.

The structure above represents the relative order observed in Korean, i.e., the head-final parameter setting (ii). In English, the setting of this parameter is (i). This ordering information is encoded in the grammar network by virtue of the relative ordering of integer id's associated with network links. Other types of parameters encoded in the grammar network are those pertaining to basic categories (i.e., possible replacements for X in (1) above), pre-terminal categories (e.g., determiner), potential specifiers, and adjuncts for each basic category.

3.3.2 Trace Theory

In general, adjunct nodes are considered to be barriers to movement. However, Korean allows the head noun of a relative clause to be construed with the empty category across more than one intervening adjunct node (CP), as shown in the following:

(3) [$_{\text{CP}}$ [$_{\text{CP}}$ t_1 t_2 kyengyengha-ten] hoysa$_2$-ka manghayperi-n] Bill$_1$-un
yocum uykisochimhay issta
[$_{\text{CP}}$ [$_{\text{CP}}$ managed-Rel] company-Nom is bankrupt-Rel] Bill-Top
these days depressed is
'Bill, who is such a person that the company he was managing has been bankrupt, is depressed these days'

The subject NP 'Bill' is coindexed with the trace in the more deeply embedded relative clause. If we assume, following (Chomsky, 1986b), that relative clause formation involves movement from an inner clause into an outer subject position, then the grammaticality of the above example suggests that the Trace theory must be parameterized so that crossing more than one barrier is allowed in Korean. Our formulation of this parametric distinction is as follows:

(4) **Barriers:** (i) only one crossing permitted; (ii) more than one crossing permitted.

In English the setting be (i); in Korean the setting would be (ii).

3.3.3 Case Theory

In general, it is assumed that the relation between a case assigner and a case assignee is biunique. However, this assumption rules out so-called multiple subject constructions, which are commonly used in Korean:

(5) John-i phal-i pwureciessta
-Nom arm-Nom was broken
'John is in the situation that his arm has been broken'

The grammaticality of the above example suggests that Nominative Case in Korean must be assigned by something other than tensed Infl. Thus, we parameterize Case Assignment as follows:

(6) **Case Assignment:** Accusative case is assigned by transitive V; Nominative case is assigned by (i) tensed Infl; (ii) IP predication.

In a biunique case-assignment language such as English, the setting for Nominative case assignment would be (i); in Korean, the setting would be (i) and (ii).

3.3.4 Summary of Parameter Settings for English and Korean

We have just seen that certain types of syntactic parameterization may be captured in the grammar network (e.g., $\overline{\mathrm{X}}$ parameters such as constituent order). In addition to these, there are syntactic parameters (e.g., Trace and Case) that must be programmed into the message-passing mechanism itself, not just into the grammar network.

We adopt the syntactic parameters of (Dorr, 1993a) and focus on the automatic generation of the corresponding grammar networks from $\overline{\mathrm{X}}$ parameter settings (i.e., Basic Categories, Pre-terminals, Constituent Order, Specifiers, and Adjunction). The parameter compilation algorithm consists of two steps: the first defines the basic structural description (i.e., bar-level nodes) using the Basic Categories, Constituent Order, and Pre-terminals parameters; and the second defines specifier and adjunct links using the Specifiers and Adjunction parameters. The English and Korean grammar networks in figure 1 are the result of executing this algorithm on the Korean $\overline{\mathrm{X}}$ parameter settings.

3.4 Results of Time Test Comparisons

As a broad-coverage system, PRINCITRAN is very efficient. The parsing component (PRINCIPAR) processes real world sentences of length 20 to 30 words from sources such as Wall Street Journal within a couple of seconds. The complexity of the current version of the system has not yet been formally determined. However, we have demonstrated that the efficiency of the system is not purely a result of using an efficient programming language (C++); this has been achieved by running experiments that compare the performance of the parser with two alternative CFG parsers. Since PRINCIPAR has a much broader coverage than these alternative approaches, the absolute measurements do not provide a complete picture of how these three systems compare. However, the most interesting point is that the trends of the three performance levels relative to sentence length are essentially the same. If PRINCIPAR had an **average** case complexity that was exponential relative to sentence length, but had only managed to be efficient because of the implementation language, the sentence length vs. performance curve would clearly be different from the curves for CFG parser which are known to have a **worst** case complexity that is polynomial relative to sentence length.

The two CFG parsers used for comparison are: a C implementation of Tomita's parser by Mark Hopkins (University of Wisconsin-Milwaukee, 1993). The test sentences are from (Tomita, 1986). There are 40 of them. The sentence lengths vary from 1 to 34 words with an average of 15.18. Both CFG parsers use the Grammar III in (Tomita, 1986, p.172–6), which contains 220 rules, and small lexicon containing only the words that appear in the test sentences. The lexicon in PRINCIPAR, on the other hand, contains about 90,000 entries extracted from machine readable dictionaries.

Tomita's parser runs about 10 times faster than PRINCIPAR; Lin and Goebel runs about twice as fast.[6] To make the parsing time vs. sentence length distribution of these three parsers more comparable, we normalized the curves; the parsing time of each of the CFG parses was multiplied by a constant so that they would have the same average time as PRINCIPAR. The adjusted timings are plotted in Figure 4. These results show that PRINCIPAR compares quite well with both CFG parsers.

3.5 Implications for Machine Translation

Our ultimate objective is to incorporate the parameterized parser into an interlingual MT system. The current framework is well-suited to an interlingual design since the linking rules between the

---

[6] This is much slower than the result reported in (Lin and Goebel, 1993). This is because we are using a simulated version of the original system (i.e., PRINCIPAR without attribute values in the grammar and lexicon). The overhead for operations such as memory allocation/deallocation is higher in the simulated version, despite the fact that attribute values are not used.

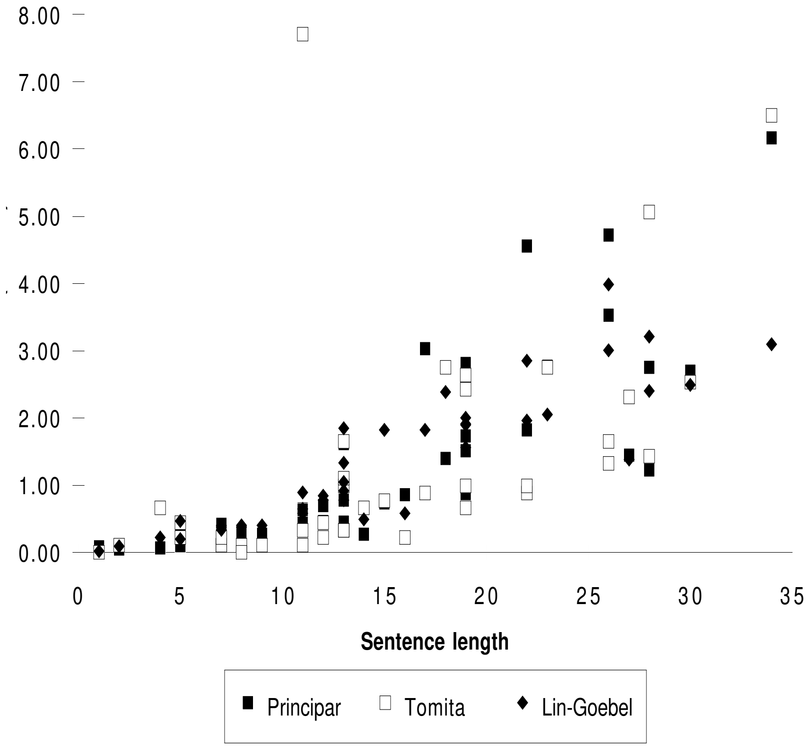


Figure 4: Adjusted Timings of Three Parsers

syntactic representations given above and the underlying lexical-semantic representation are well-defined (Dorr, 1993b). We adopt the Lexical Conceptual Structure (LCS) of Dorr's work and use a parameter-setting approach to handle well known, translationally divergent sentences (Dorr, 1990).

A parametric approach to mapping between the interlingua and the syntactic structure in English, Spanish, and German is described by (Dorr, 1990), (Dorr, 1993a). We present analogous examples here for English and Korean:[7]

(7) **Structural Divergence:**
E: John married Sally
K: John-i Sally-wa kyelhonhayssta
-Nom -with married
'John married with Sally'

(8) **Conflational Divergence:**
E: John helped Bill
K: John-i Bill-eykey towum-ul cwuessta
-Nom -Dative help-Acc gave
'John gave help to Bill'

(9) **Categorial Divergence:**
E: John is fond of music
K: John-un umak-ul coahanta

---

[7] The examples presented here are not necessarily geared toward demonstrating the full capability of the parser, which handles many types of syntactic phenomena including complex movement types. (See (Lin, 1993) for more details.) Rather, these examples are intended to illustrate that the parser is able to handle translationally contrastive sentences equally efficiently.

Parse		Time
E:	[CP [Cbar [IP [NP [Nbar [N John]]] [Ibar [VP [Vbar [V_NP married] [NP [Nbar [N Sally]]]]]]]]]	.15 sec.
K:	[CP [Cbar [IP [NP [Nbar [N John-i]]] [Ibar [VP [Vbar [PP [Pbar [NP [Nbar [N Sally]]] [P wa]]] [V_PP kyelhonhaysssta]]]]]]]	.12 sec.
E:	[CP [Cbar [IP [NP [Nbar [N John]]] [Ibar [VP [Vbar [V_NP helped] [NP [Nbar [N Bill]]]]]]]]]	.10 sec.
K:	[CP [Cbar [IP [NP [Nbar [N John-i]]] [Ibar [VP [Vbar [PP [Pbar [NP [Nbar [N Bill]]] [P eykey]]] [NP [Nbar [N towum-ul]]] [V_PP_NP cwuessta]]]]]]]	.19 sec.
E:	[CP [Cbar [IP [NP [Nbar [N John-un]]] [Ibar [VP [Vbar [V_AP is] [AP [Abar [A fond] [PP [Pbar [P of] [NP [Nbar [N music]]]]]]]]]]]]]	.12 sec.
K:	[CP [NP [0] [Nbar [N John-un]]] [Cbar [IP t [0] [Ibar [VP [Vbar [NP [Nbar [N umak-ul]]] [V_NP coahanta]]]]]]]	.07 sec.

Figure 5: Parameterized Parsing of English and Korean Divergence Examples

-Nom music-Acc like
'It is John (who) likes music'

We ran the parameterized parser on both the English and Korean sentences shown here. The results shown in Figure 5 were obtained from running the program on a Sparcstation ELC. In general, the times demonstrate a speedup of 2 to 3 orders of magnitude over previous principle-based parsers on analogous examples such as those given in (Dorr, 1993b). Even more significant is the negligible difference in processing time between the two languages, despite radical differences in structure, particularly with respect to head-complement positioning. This is an improvement over previous parameterized approaches in which cross-linguistic divergences frequently induced timing discrepancies of 1-2 orders of magnitude due to the head-initial bias that underlies most parsing designs.

A preliminary investigation has indicated that the message-passing paradigm is useful for generation as well as parsing, thus providing a suitable framework for bidirectional translation. The algorithm for generation is similar to that of parsing in that both construct a syntactic parse tree over an unstructured or partially structured set of lexical items. The difference is characterized as follows: in parsing, the inputs are sequences of words and the output is a structure produced by combining two adjacent trees into a single tree at each processing step;[8] in generation, the inputs are a set of unordered words with dependency relationships derived from the interlingua (LCS). The generation algorithm must produce structures that satisfy the same set of principles and constraints as the parsing algorithm.

Three areas of future work are relevant to the current framework: (1) scaling up the Korean dictionary, which currently has only a handful of entries for testing purposes;[9] (2) the installation of a Kimmo-based processor for handling Korean morphology;[10] and (3) the incorporation of non-structural parameterization (i.e., parameters not pertaining to $\overline{\text{X}}$ theory such as Barriers and Case

---

[8] The LCS composition routine described in (Dorr, 1992) derives the interlingua from the resulting syntactic representation.

[9] Our English dictionary has 90K entries, constructed automatically by applying a set of conversion routines to OALD entries. We have begun negotiations with the LDC for the acquisition of a Korean MRD for which we intend to construct similar routines.

[10] The English dictionary used by the message-passing system contains all morphological derivatives of every word. This approach would be impractical for Korean since the morphology is significantly richer.

Assignment).

3.6 LCS Composition

We are in the process of building the routines that convert the output of the parser described above into the LCS representation. The specification for the parser output is given in the form of a BNF in Appendix B. Sample syntactic and LCS dictionaries for Korean are given in Appendix C. Sample parse-tree and LCS outputs for a set of Korean and English sentences are given in Appendix D.

One of the advantages to the LCS representation is that it is easily extensible to a speaker-hearer model of discourse, as has been demonstrated by Dr. Martha Palmer and her NLP team at the University of Pennsylvania. A preliminary set of discourse structures for military messages is given below; these are samples that will ultimately be built automatically in the next phase of this project.

1. Message from "ops" to "bde"

   Text: COMMO IS UP.

   LCS:

```
[event CAUSE
        [thing SPEAKER]
        [state BE-ident
          [thing HEARER]
          [position AT-ident
            [property KNOW-IS-TRUE
              [state BE-ident
                [thing RADIO]
                [position AT-ident [property FUNCTIONAL]]
                [time NOW]
              ]
            ]
          ]
          [time [NOW to PLUS-INFINITY]]
        ]
        [time NOW]
      ]
```

2. Message from "ops" to "bde"

   Text: PLEASE SEND A CURRENT CMDRS REPORT ON ALL UNITS.

   LCS:

```
[event REQUEST
       [thing SPEAKER]
       [event CAUSE
         [thing HEARER]
         [event GO-poss
           [thing CURRENT-CMDRS-REPORT-ON-ALL-UNITS]
           [path TO-poss [position AT-poss [thing SPEAKER]]]
           [time [NOW to PLUS-INFINITY]]
         ]
         [time [NOW to PLUS-INFINITY]]
```

```
          ]
          [time NOW]
        ]
```

3. Message from "ops" to "bde"

   Text: DID YOU RECEIVE OUR LAST REQUEST?

   LCS:

```
[event REQUEST
       [thing SPEAKER]
       [event CAUSE
         [thing HEARER]
         [state BE-ident
           [thing SPEAKER]
           [position AT-ident
             [property KNOW-IS-TRUE
               [event GO-poss
                 [thing NOMINALIZATION
                   [event REQUEST'
                     [thing SPEAKER]
                     [event CAUSE' [thing X] [state/event Y]]
                     [time Z]
                   ]
                 ]
                 [path TO-poss [position AT-poss [thing HEARER]]]
                 [time [MINUS-INFINITY to NOW]]
               ]
             ]
           ]
           [time NOW]
         ]
         [time NOW]
       ]
       [time NOW]
     ]
```

4 Translation Issues

Upon completion of the previous annual report (January, 1994), we investigated certain issues that were left unresolved with respect to the more difficult translations between Korean and English. Some of these issues are addressed here.

4.1 Paraphrases and Resumptive Pronouns

In the previous report, some of the translations seemed to be paraphrases. An example of such a case is the following:

```
(10) [cp[cp e1 e2  kyengyengha-ten]  hoysa2-ka  manghayperi-n]
                          managed-Rel  company-Nom is bankrupt-Rel
     Bill1-un  yocum        uykisochimhay issta
```

-Top these days is depressed
'Bill is such a person that the company which was managed
by him has been bankrupt, and he is depressed these days'

Here, the gloss "Bill is such a person that the company which was managed by him has been bankrupt, and he is depressed these days" is the closest translation that we were able to provide. Note that one might mistakenly attempt to translate this as: "Bill, who managed the company which was bankrupt, is depressed these days" since this is a much more natural translation. This would be incorrect, however.

Consider these two cases more carefully:

(11) [[t2 t1 kyengyengha-ten] hoysa1-ka   mangha -n]   Bill2-un...
managed-Rel company-Nom was bankrupt-Rel Bill-Top
(Literally): 'Bill2, (who2) the company1 (which1) HE2 was managing
has been bankrupt, ...'

(12) [t2 [t1 mangha-n]   hoysa1-lul kyengyenghayss-ten] Bill2-un...
was bankrupt-Rel company-Acc   managed -Rel   Bill-Top
'Bill2, who2 managed the company1 which1 was bankrupt,...'

In both of these there are two relative clauses, one embedded inside the other. Also, in both cases, the more deeply embedded relative clause is modifying 'hoysa'(company) and the other one is modifying 'Bill'. However, there is a crucial difference between these two sentences. In the first one, it is Bill who is bankrupt. In the second one, it is the company that is bankrupt. Note that in the second case, 'Bill' can be associated with its trace (=t2) across only one S boundary. However, 'Bill' in (17) cannot be associated with its trace in that way. Rather, it is associated with its trace across two S boundaries.

In this case, it seems there is no direct English translation: associating a trace with its antecedent across more than one bounding node results in ungrammaticality in English (given that the trace results from a single application of movement). In a later version of this, we changed the translation to be less awkward, using the pronoun "whose" and a resumptive pronoun (as suggested by Yaeger):

Bill, whose company he was managing is bankrupt [so he] is depressed
these days.

However, given the device we are using for parsing (described in the last section), it seems there is no computational inexpensive way of restructuring the phrase so that it uses the resumptive pronoun. Such a modification would require the repositioning of traces, which is heavily computation-intensive.

4.2 Use of Chinese Characters for Semantic Distinctions

In the previous report, the following example was presented as a divergence:

(13) K: John-un   khi-ka   khuta
-Top height-Nom   is big/huge
'John is tall'
E: John is tall

In translating, there are important semantic distinctions between Korean words and their Chinese character counterparts. In the above example, the Korean word "khuta" is a predicate adjective that stands on its own. It would not be possible to use the Chinese analogue, "tay" because the latter is exclusively used as a modifier of a noun.

Often it is difficult to distinguish between the two cases:

(14) tay-cicin / cicin
big-earthquake earthquake

(15) tay-hak / *hak
big-study(?) study

In the first example above, 'tay' is a modifier of 'cicin'(earthquake). Crucially, 'cicin' can be used without 'tay' (as an independent word). This is not the case in the second example. The word 'tay-hak', which means 'university', cannot be separated as 'tay'+'hak'. (In other words, 'hak' alone cannot be used for representing 'study'.) In this case, 'tay' seems to be a word in its own right, i.e., an idiomatic meaning results. We may simply distinguish between such by considering 'tay' to be a modifier in the former but not in the latter.

On the other hand, the Korean word 'khu-'(big) can be used not only as a modifier but also as a predicate. In the former usage, it combines with adnominalizer '-n', resulting in 'khu-n'. In the latter usage, it combines with declarative marker'-ta' (or interrogative marker '-ni'), resulting in 'khu-ta' (or 'khu-ni').

4.3 Syntactic Structure of Korean

During the course of our study of the structure of Korean, we have attempted to address the issue of the necessity of a CP (clausal) node. The issue of whether Korean has a CP has been very controversial. We claim that a CP is, indeed, necessary and that there exist several empirical and conceptual arguments to support this.

The first argument is related to the notion of barrierhood. In (Chomsky, 1986a), the CP node can be a barrier, but IP cannot be a barrier by itself. IP can be a barrier only by inheritance from the maximal projection(=XP) it is dominating. If we hypothesize that there is no CP in Korean, IP must be a barrier, given that (at least) some movement out of certain clause creates ungrammaticality. This in turn leads to the (unmotivated) asymmetry between English and Korean: English IP is not a barrier by itself while Korean IP is. But if we assume that there is CP in Korean, then we need not resort to such an unmotivated asymmetry.

In Korean extraction phenomena in certain constructions (like the topic construction) are not accounted for by Subjacency. However, we still seem to need the notion of barrierhood. So, we adopt the assumption that there is a CP node in Korean and that non-L-marked CP is a barrier.

Another argument for the necessity of CP comes from the structure of relative clause or adverbial clause.

```
(16) [ t1   Mary-lul coaha-nun] salam1
            Mary-Acc  like-Rel   man
       'the man who likes Mary'
```

If there is no CP in Korean, then there is no site for accommodating the relative clause marker '-nun'; it cannot be placed in the Spec of IP or IP adjoined position. Since a relative clause should be distinguished from a complement clause, for instance, we need a place for accommodating a category which signals the type of clause, i.e., complementizer, relative clause marker, adverbial clause marker etc. This means that we must have CP (above IP).

4.4 Scrambling as Case-Driven Obligatory Movement

Another area that we have investigated intensively since the previous report is the issue of scrambling. In (Lee, 1993), movement is not optional, but is a consequence of case-driven obligatory movement. Some points about her proposal are the following:

(A) Scrambling is adjunction.

(B) Scrambling is A-movement: It exhibits properties of A-movement (in terms of Binding and Weak Crossover phenomena).

(C) Consequently, adjoined positions are A-positions in Korean.

(D) Scrambling is a consequence of case-driven movement. (Assuming the VP-internal subject hypothesis, all the arguments move out of VP to be assigned case. As long as case licensing conditions are met, arguments may be arranged in any order.)

There are some consequences and implications of the above proposal. The first is that Theta-role assignment is completely dissociated from case assignment. The second is that even a canonical word order sentence involves obligatory movement (scrambling). Finally, this proposal predicts a a parametric difference between English and Korean. The difference is the level at which accusative case is assigned: In Korean, accusative case is licensed at S-structure, while in English it is done at LF. (Such a difference is reduced to the level at which verb raising to INFL takes place.)

We adopt this proposal and take it one step further by proposing an algorithm that will allow us to bind empty arguments corresponding to scrambled elements. In particular, if an argument is scrambled out of a clause, we can re-claim the argument inside of clause by using a trace or a crossing branch:

```
(17) John said that Mary likes Tom.
     John-top Mary-nom Tom-acc like+comp said. (basic order)
     Tom-acc John-top Mary-nom like+comp said. (scrambled, Tom is outside)
```

If it's coindexed, we can recover the argument by having a trace or by allowing more than one terminal point to the same NP:

```
(18) John said that () likes Tom.
     John-top Mary-nom like+comp said.

     Analysis 1:
     John-top1 e1 Mary-nom like+comp said.

     Analysis 2:
     e1 John-top1 Mary-nom like+comp said.
```

If there is no preceding NP with the case required by a particular verb, then topic-marked NP will be converted to an accusative-marked NP:

```
(19) John said that Mary likes ().
     John-top Mary-nom (here need NP-acc) like+comp said.

     Analysis: John-top1 Mary-nom e1 like+comp said.
```

Note: John-top1 is co-indexed with e1 which need an accusative NP antecedent.

We are in the initial stages of designing an algorithm that takes the above points into consideration. This will be installed in the parser during the next phase of the project.

4.5 Empty Categories

As part of our work to reduce the complexity of movement and scrambling, we have had to face the difficulty of recoverability of Empty Categories (EC's). On approach that has been suggested is that of (Egedi et al., 1994), in which the general rule that is used is: 'Choose the closest topic that matches the semantic constraints on the elided arguments'. This approach is computationally efficient in that the antecedent of EC is recovered without delaying the actual computation of sentence structure. However, there is a problem with rule. In particular, there are frequently cases where the closest topic cannot be the antecedent of the EC, as seen in the following:

```
(20) Mary-ka  [John-i  e  hyeppakhayssta-ko]  malhayssta
     Mary-Nom  John-Nom   threatened-Comp       said
     'Mary said that John threatened her.'
```

The algorithm is to search through the list for each dropped argument, matching the semantic feature constraints from the verb against the features on the NP until the appropriate NP is found. Roughly, the structure assumed for the above sentence (with English words inserted) would be:

```
            S
          /   \
     Mary_1     VP
              /   \
            S    said
          /  \
     John_2   VP
             / \
            e   threatened
```

In this example, the empty object in the embedded clause must be coreferential with the matrix subject (Mary), although the closer NP to the empty object is the embedded subject (John). (cf: In their paper, only topic NPs have been discussed. However, their general rule can be (and must be) applicable to NPs other than topic-marked ones. In (1), there is no topic-marked NP; thus, the antecedent of EC should be sought among non-topic NPs.) In fact, as discussed in (Suh, 1994), empty objects in Korean tend to be bound by an NP outside of the current clause probably because they are pronominals constrained by the Binding Condition B. Such a grammatical principle should be implemented in in order to make correct predictions.

Note that one could argue that there is a semantic property associated with "threaten" (i.e., that one cannot threaten oneself) that forces the correct binding to fall out. However, it is easy to come up with a counterexample. Consider the following structure, which is legal in Korean:

```
            S
          / \
     Mary_1 VP
             / \
            S said
          / \
     John_2 VP
             / \
            e liked
```

The algorithm above would obtain an incorrect result, i.e., the empty category, e, would be incorrectly bound to the closest NP John, yet it should be bound to the matrix subject NP Mary. Note that it is possible for someone to "like himself". In particular, it is not the case that what is responsible for the binding of EC from outside of the current clause is a semantic factor.

Consider another example in which it is semantically possible for the EC to be bound by the closest NP, i.e., the EC still should be bound from outside of the current clause:

```
(21) John-i [Mary-ka e piphanhayssta-ko/ chingchanhayssta-ko]  malhayssta
     John-Nom Mary-Nom EC criticized-Comp/ praised-Comp         said
     'John said that Mary criticized/ praised  him (or someone else).'
```

It is semantically possible for someone to criticize or praise himself/herself. But the example above does not allow such a semantic interpretation, and the EC should find its antecedent outside of the current clause. Consequently, 'John' (or somebody else available from the context) is coindexed with the EC. It seems clear from this example that such an EC is a pronominal subject to Binding Condition B (or a variable subject to Binding Condition C).

5 Temporal/Aspectual Analysis

In the next phase of the project, we will be collaborating with Dr. Mari Olsen from Northwestern University in order to install temporal/aspectual knowledge into our semantic representations. As a preliminary study, in preparation for this upcoming collaboration, we have worked on a definition of basic tense structures for Korean which we expect to tie into the work of (Olsen, 1994). In addition, we have undertaken the task of investigating a number of aspectual paradigms and have subsequently attempted to distinguish among different levels of representation that are affected by aspectual information. This "layered" approach is one that will mesh well with the work of Dr. Olsen, in which temporal, lexical, and grammatical aspect are carefully separated into different tiers.

5.1 Possible Tenses in English and Korean

According to (Hornstein, 1990), in all of the languages of the world, there are (logically) 24 possible basic tense structures (BTS):

```
Present: S,R,E  S,E,R  R,S,E  R,E,S  E,S,R  E,R,S
Past:    E,R_S  R,E_S
Future:  S_R,E  S_E,R
Present perfect: E_S,R  E_R,S
Past perfect:    E_R_S
Future perfect:  S_E_R  S,E_R  E_S_R  E,S_R
Distant future:  S_R_E
Future in past:  R_S,E  R_E,S  R_S_E  R_E_S
Proximate future:S,R_E  R,S_E
```

BTSs are composed of an SR relation and an RE relation: SR and RE are either associated or separated, and are strongly ordered linearly. However, S and E are not related to each other in these ways except derivatively. Given this, we can cut down the number of possible tenses to 16:

```
S,E,R = *E,S,R
R,E,S = *R,S,E
S_E_R = *S,E_R = *E_S_R = *E,S_R
R_S,E = *R_E,S = *R_S_E
S,E_R = *E,S_R
```

Using the framework above, we developed a scheme in which English and Korean morphemes were associated with BTS representations. We will look at each of these languages, in turn:

(A) English:

- (a) i. present morpheme: associate S and R: `S,R`
    - ii. past morpheme: R removed to left of S: `R_S`
    - iii. future morpheme: R removed to right of S: `S_R`
- (b) i. +have: E removed to left of R: `E_R`
    - ii. -have: E and R associated: `E,R` or `R,E`

**There is no specific morpheme that signals the association of R and E.

(B) Korean:

- (a) i. present morpheme: associate S and R: `S,R`
    - ii. past morpheme: R removed to left of S: `R_S`
    - iii. future morpheme: R removed to right of S: `S_R`
- (b) E and R are generally associated(=E,R or R,E)

**In a given BTS, if linear order is not intrinsically determined, assume that the linear order of RE is identical to the linear order of SR. Also, morphemes unambiguously determine unique mappings.

In the final inventory of 8 tenses (shown here), Korean only has the first three. This is because E and R are generally associated (unless specified otherwise).

```
Present: (S,R)o(R,E) = S,R,E  (i)
         (R,S)o(E,R) = E,R,S  (ii)
Past:    (R_S)o(E,R) = E,R_S
Future:  (S_R)o,(R,E) =  S_R,E
Present perfect: (S,R)o(E_R) = E_S,R  (i)
                 (R,S)o(E_R) = E_R,S  (ii)
Past perfect:    (R_S)o(E_R) = E_R_S
Future perfect:  (S_R)o(E_R)
Future in past:  (R_S)o(R_E)
Proximate future:(S,R)o(R_E) = S,R_E  (i)
                 (R,S)o(R_E) = R,S_E  (ii)
```

We now consider some examples. The first case is Past vs. Past Perfect. The latter is is obtained by "reduplication" of the past tense marker:

```
(22) John-un Mary-lul  piphanhay-ss-ta
         -Top    -Acc  criticize-Pst-Ind
     'John criticized Mary.'
          E,R_S
```

```
(23) John-un Mary-lul piphanhay-ss-ess-ta
         -Top    -Acc criticize-Pst-Pst-Ind
     'John had criticized Mary.'
     E_R_S
```

The second case is Future vs. Future Perfect. The latter interpretation is available when the topic marker is attached to the temporal adverbial phrase, which amounts to the R point:

```
(24) John-un  han     si-ey  swukcey-lul kkutnay-l kes-ita
         -Top one o'clock-at homework-Acc   finish-will
     'John will finish his homework at 1 p.m.'
     S_R,E
        |
        1 p.m.
```

```
(25) John-un  han     si-ey-nun  swukcey-lul kkutnay-l kes-ita
         -Top one o'clock-at-Top homework-Acc  finish-will
     'John will finish his homework by 1 o'clock.'
     'John will have finished his homework at 1 o'clock.'
     S_E_R
         |
       1 p.m.
```

Finally, consider the case of Present perfect vs. Past. Present perfect meaning is represented through the use of particular expressions involving (auxiliary) verbs.

```
(26) John-un New York-ulo  ka pely-ess-ta/    ka-ko    ep-ta
         -Top         -to   go  Aux-Pst-Ind   go-conj  not exist-Ind
     'John has gone to New York.'
     E_S,R
```

```
(27) John-un New York-ulo  ka-ss-ta
         -Top         -to   go-Pst-Ind
     'John went to New York.'
     E,R_S
```

Due to the lack of a grammatical device for representing perfect tense (corresponding to English auxiliary 'have'), emphasizing the R point seems necessary in Korean in order to represent perfective meaning explicitly. Case (27) is such an instance, where the topic marker is attached (for making the R point salient) to the PP referring to the R point. (Such an usage of topic marker is also observed in past perfect.)

5.2 The Place of Aspect in an IL-Based MT System

In an effort to place our work in the wide range of analyses available in the literature we have adopted the following definitions of what "aspect" is:

(a) semantic vs. grammatical aspect:
"aspect" may appear in sentences lexically and morphologically as overt grammatical evidence for the underlying or semantic notion.

(b) situation vs. perspective aspect:
"aspect" may refer to the internal structure of a situation (states, processes, events/transitions) or to how much of the full situation is being focused on (perfective, imperfective)

(c) temporal vs. agentive/volitional aspect: "aspect" has been taken to be a temporal property by some whereas others have defined it more broadly to include agentive or volitional properties

With respect (b), we represent situation aspect using a temporal property definition of aspect. We are focusing specifically on those properties that are derived from spatial information. (For example, "John ran in the park" conveys a Process, and so does "John ran through the park". However, "John ran to the park" does not, it conveys a bounded Event.) This gives us the opportunity to test the robustness of our spatial representations and to start building a micro-theory of aspect for translation.

With respect to (a), any MT system must be able to capture both semantic and grammatical aspect. The mapping between semantic and surface or grammatical aspect is language-specific, whereas the semantic aspect itself in an IL-based MT system must be language-independent. Clearly the development of the mapping and the representation itself are related. The representation for aspect may either be the primary focus with the mapping developed secondarily, or the mapping may be primary, in which case it may drive the representation of aspect. We have chosen the first of these two approaches.

At the IL level of representation of a sentence we are translating, the MT system has access to two classes of information:

(1) language-specific information used to map from the source language lexicon entries and parse tree to the IL, and from the IL to target language lexicon entries and generation tree

(2) language-independent, IL-theory-internal definitions to map IL primitives into some subset of primitives in knowledge representation (KR) system

Though strictly speaking we take "aspect" to be a property of IL structures of type SITUATION (as opposed to type THING, SPACE, etc.), the information that goes into deriving situation-aspect, call it "i-aspect", may appear at any level in the MT system.

Any MT system that claims to be "interlingua-based" must of course justify which levels exist in the design and what constitutes evidence for the representational information at each level. As evident above, we have an IL and a KR level of representation. It is our working assumption that i-aspectual information corresponds (possibly through mappings between levels) to primitives at the KR level.

We can identify 3 ways in which i-aspectual information arrives in an IL structure. First, if i-aspectual information originates (i.e. is available) lexically or morphologically, then the MT system has access to it by virtue of it being "coded" in the system SL or TL lexicon. For example, the word "repeatedly" requires including i-aspectual information in its lexicon entry.

Second, i-aspectual information may be part of the definition of an IL primitive. The information does not appear in the SL or TL lexicons directly, only indirectly via the primitive. For example, the word "through", as in "John ran through the park", has the IL primitive VIA in its lexicon entry. It is only in the IL-theory-internal definition of VIA that we have i-aspectual information in terms of KR primitives for VIA, and hence for "through". So the MT system has access to it by virtue of it being "coded" in the IL-theory-internal lexicon, where IL primitives are listed as entries together with their KR definition.

Third, i-aspectual information may need to be derived because it is not available in either of the above ways even though the sentence being translated is considered grammatical. The derivation is contingent on the existence of "coercion" or "construal" rules that are invoked to complete an IL structure. For example, consider the sentence "she lives through the tunnel". This means she lives at the other end of the tunnel. The issue here is where does this particular interpretation of the sentence come from given that the sentence conveys a STATE situation even though the prepositional phrase contains PATH information, rather than the needed PLACE information. We could fix this lexically and define one meaning of "through" to be an augmentation of its PATH meaning (with its VIA primitive, as above) into a PLACE meaning, with an IL primitive such as AT-END-OF. (One limitation to this approach is that it may then overgenerate interpretations of "through" in cases where this meaning is not applicable.) Alternately, we could craft a construal rule that allows for such an augmentation more generally when a STATE verb requires a PLACE argument that converts a PATH into a PLACE. This is a context-dependent construal that may be more be finely-tuned than the lexical approach.

5.3 Lexical and Grammatical Aspect: Olsen

Our work with Dr. Mari Olsen has allowed us to fine-tune our specification of the levels of aspectual knowledge described above. Over the next phase of the project, we will be applying Olsen's model to MT for incorporation of aspectual knowledge into the interlingua. Olsen's privative analysis of aspect and tense allows temporal semantics to be built up monotonically, from features associated

Automation Spectrum	**Lexicon Tool**
Manual	LCS Editor
Semi-Automatic	TEAM (Grosz et al., 1987) IRUS (Bates and Bobrow, 1983) TELI (Ballard and Stumberger, 1986) LUKE (Knight, 1991) User's LCS Editor Others: (Ginsparg, 1983), (Guida and Tasso, 1983), (Grishman, 1986), (Thompson and Thompson, 1983), (Templeton and Burger, 1983), etc.
Automatic	LEXICALL

Figure 6: Spectrum of Possibilities: Lexicon Building for NLP Applications

with the verb, to those marked by other lexical and grammatical constituents and the pragmatic context (see (Olsen, 1994)).

In conjunction with this work Dr. Olsen will be collaborating with the University of Maryland on how lexical aspect may be used to improve lexical choice in machine translation. For example, Spanish has many motion verbs in which the path of the motion event is encoded (e.g. entrar 'enter'); English, according to (Talmy, 1978), (Talmy, 1991), more frequently uses verbs 'lexicalizing', i.e. expressing, the manner of the motion rather than the path (e.g. run, roll). These generalizations may translate into a difference in the frequency of telic verbs: Spanish may have more [+telic] verbs (types or tokens) than English. In some cases, therefore Spanish to English translation may be described as a mapping of a Spanish [+telic] verb (such as entrar) onto an English atelic verb with a [+telic] complement (such as go in). However complexities in this mapping need to be explored before an algorithmic description is possible.

6 Lexicon Development and Semantic Classification of Verbs

In this section, we discuss an area related to the problem of lexical acquisition, i.e., semantic classification of verbs and implementation of tools for the development of lexicons based on this classification.

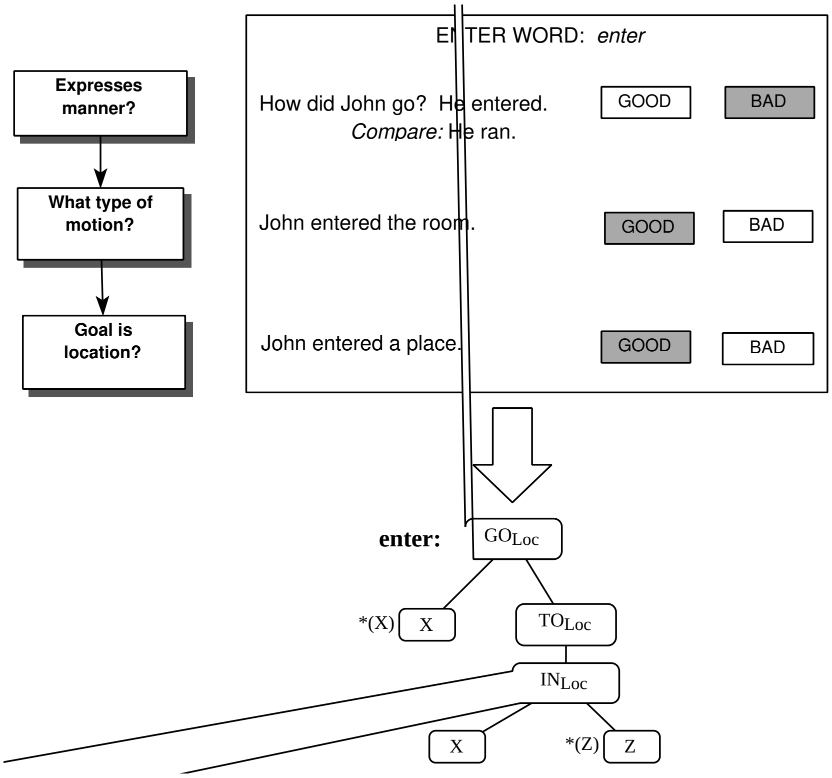
6.1 Semi-Automatic Tools for Lexicon Development

In the previous annual report, we described our LCS Editor, which provides an interface that allows LCS's to be built and modified. During editing, the current state of the LCS is continuously updated and displayed in two windows.

It is clear that, while the LCS Editor is useful, it still does not eliminate many of the aspects that lead to tedium in the construction of a dictionary. In this section, we discuss a progression of dictionary-construction ideas ranging from the LCS Editor that we've already built to a more advanced lexicon development tool. Figure 6 illustrates this range.

At one end of the lexicon-tool spectrum (Manual), the level of knowledge assumed by the dictionary builder must be relatively extensive, the level of tedium is high, and development time is lengthy; however, linguistic coverage is easily controlled by the builder (i.e., full coverage within a domain is well within the realm of possibility). At the other end of the spectrum (Automatic), there need not be any linguistic knowledge and development is automatic (i.e., not tedious or lengthy); however, there lexicon coverage is likely to have "holes," i.e., a lack of application-specific information due to the fact that the content of machine-readable resources cannot always be predicted in advance.

The use of the LCS Editor requires a great deal of linguistic knowledge because LCS's are more sophisticated than other types of knowledge that are generally encoded in NLP lexicons.



How did John go? He entered.
Compare: He ran.

John entered the room.

John entered a place.

Figure 7: User's LCS Editor

However, the editor does provide certain types of automatic constraint satisfaction that fills in certain types of nodes while the user is building the representation. This is where the benefit of the LCS representation is most apparent: the LCS conforms to wellformedness conditions that can be applied automatically during the construction of a lexical entry.

Many lexicon development tools fall under the category of "Semi-Automatic," i.e., they have been built for users with no specialized training in linguistics. Most tools in this category still look much like the first such interface created for the TEAM project (Grosz et al., 1987). One of the most sophisticated tools in the Semi-Automatic category is the LUKE system by (Knight, 1991). This system comprises the lexical acquisition component of large NLP system called KBNL (Barnett et al., 1990). LUKE is connected to a knowledge-representation system called CYC (Lenat and Guha, 1990), a large-scale, common-sense, knowledge-based (KB) project. The drawback of this system is that the person who does editing must be KB expert and, in particular, must know about concepts available in the CYC system.

The approach used in systems like LUKE is clearly a step in the right direction toward automation of the lexicon construction process. While many of the ideas behind these systems have been borrowed in our approach to lexicon construction (e.g., the mapping from syntactic constituents to semantic roles), we have found that it is possible to avoid certain drawbacks such as the requirement that the lexicon developer be a KB expert.

The latest version of our LCS Editor is one that allows a non-linguist i.e., a "User" of the Foreign Language Tutoring system — generally assumed to be a foreign language instructor — to enter words into the lexicon without being forced to manipulate the LCS representation directly. (See figure 7.) This tool does not require as much linguistic knowledge as our already existing Lexicon Tool, but is still in the semi-automatic category as is the LUKE system. This approach has the advantage that it does not require a large knowledge base (or a KB expert). However, as in LUKE, the system presents easy-to-read sentences and the user is then asked to assign grammaticality judgements to these sentences. Note that the menu from the previous figure is given in parallel here to illustrate the correspondence between menu-driven questions and the relevant sentences. Once the user makes a judgement about these sentences, the information is then used to infer

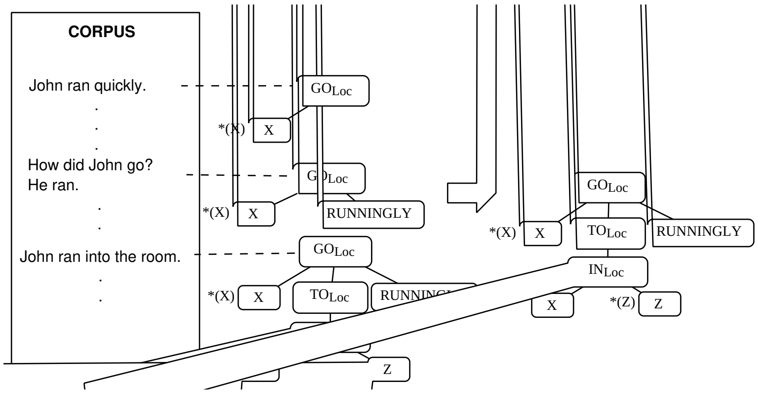


Figure 8: Toward Automatic Lexical Acquisition: Searching a corpus for sentences

information about the LCS representation.

The key to the success of this design is the use of an appropriate set of sentences that allow automatic inference information about the LCS. For example, the verb *enter* is different from *run* in that the former does not express manner of motion. This subtle distinction is brought out by asking the user to specify whether *How did John go?/He X-ed* is an appropriate sentence pair. For *run* this pair is appropriate since the "how" portion of the verb is inherent in its meaning — he went "runningly"; by contrast, for *enter* this pair is not appropriate since there is no "how" component of meaning.

If we push the above design one step closer to automated lexical acquisition, we might consider building a sentence-finding program that searches a textual corpus (i.e., a large body of sentences in a particular language) for the occurrence of sentences (or sentence pairs) that correspond to the ones in the presentation window of figure 7. Thus, instead of developing the appropriate set of sentences to isolate meaning, the task is now to devise a program that is able to select appropriate sentences from corpora. Figure 8 illustrates this modified design. In this example, the system is attempting to determine the LCS representation for the verb *run*. Note that each sentence that is found to be relevant in the corpus is analyzed for information that would add one component of meaning to the LCS representation. For example, the verb phrase *run quickly* in the first sentence indicates that this is a GO action; the addition of the manner component *runningly* can then be derived from the sentence pair, i.e., from the occurrence of a sentence that answers the question "how".

The key to the success of this type of automatic design is to use an extensive corpus (for each of the languages of interest) and a "smart" analyzer. Unfortunately, as shown by (Hogan and Levin, 1994), it is a non-trivial task to acquire even simple components of meaning, such as argument structure information, by means of syntactic extraction out of a corpus. Moreover, a corpus has inherent limitations; we have no guarantee that we will be able to find the precise sentences we need in order to predict LCS components of meaning. For example, even with a corpus as enormous as the Lancaster-Oslo-Bergen (LOB) corpus — 60,000 sentences — there is no occurrence of the construction "How did `X Verb`", with or without a subsequent response of the form "`X Verb`ed".

The difficulty of searching for *particular* constructions has led us to investigate a higher level of abstraction that utilizes the verb classification presented in (Levin, 1993). Our goal is to test Levin's hypothesis that the syntactic behavior of a word is fully semantically determined and, moreover, that this property holds across all languages. For example, the verb *run* is a manner-of-motion verb which, according to Levin, participates in the following syntactic alternations:

Path	The horse ran through/into/out of the stream
There Insertion	A horse ran out of the barn There ran out of the barn a horse
Locative Inversion	A horse ran out of the barn Out of the barn ran a horse
Measure Phrase	We ran 5 miles
Resultative Phrase	We ran ourselves into a state of exhaustion

Searching for alternations in a corpus has led to more promising results than searching for a small set of language-specific constructions. Part of the investigation of this approach has involved the compilation of a large matrix that crosses syntactic alternations (approximately 80) with semantic classes (approximately 192). Preliminary results (Corbin et al., 1994) have demonstrated that this approach is easy to implement; however, the mapping from semantic classes to LCS representations is non-trivial. Moreover, many of the syntactic alternations are specific to English, which means that we would have to build a table like this for every language for which we need a dictionary. Given that the work of (Levin, 1993) was the result of compilation of at least a decade of research articles and notes, it is clear that the construction of a table for every language is a non-trivial task.

Another concern is the danger of committing to specific alternations that might have radically different analogues in other languages.[11] (Levin, 1993) argues that when the same alternations *do* exist across languages, the same semantic classification is supported; however, (Mitamura, 1990) opposes this point of view, arguing that Japanese associates syntactic patterns with semantic classes, but that these semantic classes are not the same as those developed by Levin for English. She claims that syntactic alternations in Japanese are based primarily on case distinctions, for example, verbs associated with the *ga* and *kara* case markers are in the semantic class of *giving*. Since there is no semantic class of 'giving' of this type in (Levin, 1993), Mitamura argues that Levin's semantic classification is not correct.

In our own study of Korean (which is similar to Japanese in structure), we too have discovered that the syntactic tests for English do not apply directly to Korean verb classes. An example is the transitivity alternation, which is an important criterion for Levin's verb classification. This alternation is irrelevant to Korean because it is not possible for a Korean verb (such as *break*) to function as both a transitive and an intransitive verb. However, it is not clear that the tests that Mitamura was relying on are as systematic as they could be. In particular, Mitamura's grammatical assumption is that the notions of SUBJECT and OBJECT are primitive (as in Lexical Functional grammar). She provides the following examples of Alternations in English and Japanese:

(28) a. Brian supplied a blanket to John.
b. Brian supplied John with a blanket.

(29) a. Brian-ga John-ni mouhu-o sikyuusita
-Nom -Dat blanket-Acc supplied
'Brian supplied a blanket to John.'
b.*Brian-ga John-o mouhu-de sikyuusita
-Nom -Acc blanket-with supplied

(30) a. kodomo-ga kawa-de oyoida
child-Nom river-in swam
'A/The child swam in the river.'
b. kodomo-ga kawa-o oyoida

---

[11] Surprisingly, the analogues of many English syntactic alternations do actually exist in other languages. For example, the Australian language Warlpiri has a middle construction (i.e., *The bread cuts easily*).

```
      child-Nom  river-Acc   swam
      'A/The child swam in the river.'
```

Mitamura argues that the alternation above occurs with the movement verbs which take a locative phrase. (The marker -de indicates a general location where the movement is held, and the marker -o indicates the area covered by the movement.)

Our view is that the notions of 'subject'/'object' as primitives is highly suspect. The validity of the 'subjecthood'/'objecthood' test employed by Mitamura is questionable. Consider the following examples:

```
(31) a. John-i  Bill-ul   ttayli-ess-ta
            -Nom     -Acc     hit-Pst-Ind
        'John hit Bill.'

     b. *Bill-i  John-eyuyhay  ttaylie-ci-ess-ta
             -Nom     -by          hit-Pass-Pst-Ind
        'Bill was hit by John.'

(32) a. John-i  Bill-ul   kwuthaha-ess-ta
            -Nom     -Acc      hit-Pst-Ind
        'John hit Bill (repeatedly).'

     b. Bill-i  John-eyuyhay  kwutha-tangha-ess-ta
            -Nom     -by          hit -Pass-Pst-Ind
        'Bill was hit by John (repeatedly).'
```

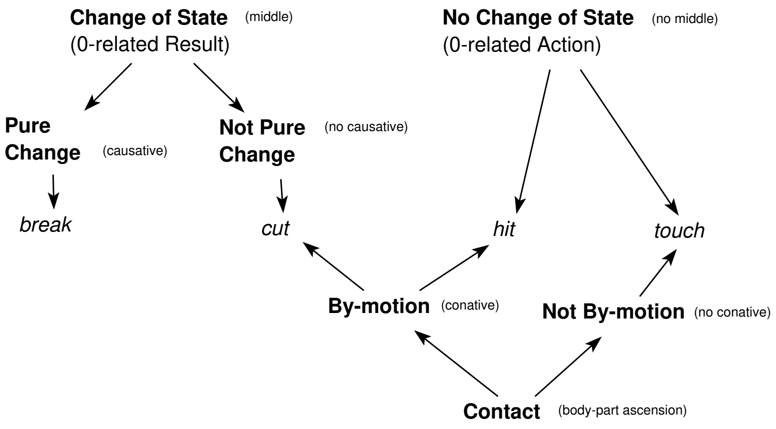
Verbs such as ttaylita(hit), chita(strike), cwukita(kill), sata(buy), yokhata(curse), mannata(meet), topta(help), chingchanhata(praise) etc. take an accusative NP as their complement, which is NOT passivizable. According to Mitamura, such an accusative NP is not an object but an oblique NP, and the verbs taking such an NP as a complement would be distinguished from the verbs (with similar meaning) which take a passivizable complement. This is not the desired result. Thus, we dismiss the notions of subject, object, and oblique (as grammatical function), and concentrate on 'typical' case marking and 'typical' theta role of NPs in classifying verbs.

Levin's work does not assume that a verb in one language necessarily behaves the same as all of its equivalent forms in another language, especially if that verb has multiple verb senses. For example, there is no single Korean counterpart for all senses of the English verb *break*, so it should not be the case that *all* syntactic behaviors associated with the word *break* carry over to *all* possible counterparts in another language. We take the view adopted by Levin that, while a verb in one language might not have an exact counterpart in another language, the "basic meaning components" that define the lexical items of each language are the same.

Our approach to automatic lexicon construction relies heavily on this notion of "basic meaning components." If we can decompose Levin's semantic classification into primitive units of meaning, then we should be able to conduct a more thorough investigation of whether the proposed semantic classification is indeed applicable to other languages.

6.2 Automated Lexicon Construction: LEXICALL

We have designed a lexical acquisition program that relies on a systematic relation between syntactic behavior and "basic meaning components" for the construction of LCS representations. We take as our starting point the 3100 verbs that have already been classified in (Levin, 1993). Through the statistical process described below in section 7, we classified an additional 3100 verbs. Once we arrived at the full Levin-based classification for 6200 verbs, our acquisition algorithm used a



Change of State -> Ident (w/ state change as result)
No Change of State -> Not Ident
Pure Change -> optional CAUSE
Not Pure Change -> obligatory CAUSE
By-Motion -> Theme/Result (without manner)
Not By-Motion -> Manner "ingly" form
Contact -> * in two positions

BREAK: [CAUSE (W, [GO-Ident (*X, [TO-Ident ([AT-Ident (X, BROKEN)])])])]
CUT: [CAUSE (W, [GO-Ident (X, [TO-Ident ([*AT-Ident (*X, CUT)])])])]
HIT: [CAUSE (W, [GO-Loc (HIT, [TO-Loc ([* AT-Loc (HIT, *Z)])])])]
TOUCH: [BE-Perc (W, [AT-Loc (W, *Z)]) TOUCHINGLY]

Figure 9: Automatic Acquisition of LCS Representations from Levin's Classification

set of language-independent mappings from the verb class of a given verb into the components of meaning associated with the LCS representation. Figure 9 illustrates this process for a small set of verbs: *break*, *cut*, *hit*, and *touch*.

An important aid for determining our "basic meaning components" was the LDOCE (Proctor, 1978), which is based on a "control" vocabulary of 2000 words (i.e., the entire inventory of words used in LDOCE definitions). Approximately 700 of these are verbs, only 114 of which do *not* appear in Levin's book; we classified these 114 verbs (by hand) in terms of Levin's semantic categories and then used the LDOCE as a guide to "learn" the basic meaning components for the 3100 verbs not occurring in (Levin, 1993) by looking at verb definitions described by the control vocabulary. We were then able to automatically derive the LCS representation (as described above) for these verbs. For example, the definition of *run* in the LDOCE is "to move on one's legs at a speed faster than walking." The verb *move* is one of the control vocabulary words; it has a semantic class defined in Levin's hierarchy. Thus, there is an associated LCS for *move*: $[_{\text{Event}}\ \text{GO}_{\text{Loc}}$ ([Thing], [Manner]*)]. The LCS that results from combining the LDOCE dictionary definition of *run* plus the LCS for *move* is the following:

$$\begin{array}{l}[_{\text{Event}}\ \text{GO}_{\text{Loc}}\\ \quad ([_{\text{Thing}}\ \text{W}],\\ \quad [_{\text{Manner}}\ \text{BY}_{\text{Instr}}\ ([\text{Thing LEG}])],\\ \quad [_{\text{Manner}}\ \text{FAST}])]\end{array}$$

One of the main contributions of the approach described here is that it provides a relation between Levin's classes and meaning components as defined in the LCS representation. For example, we are able to determine whether the verb has an Identificational component of meaning in the LCS simply by checking whether the verb is in the Change of State class. The verbs *break* and *cut* have this component of meaning in the LCS representation constructed by the acquisition procedure, whereas, *hit* and *touch* do not.

An important benefit of using the Levin classification as the basis of our LCS derivation is that, once all the LCS's for the verbs have been derived, it is not necessary to store all aspects of verb behavior in the lexical entry. In particular, general principles determine syntax (i.e., alternations) from semantics. In the XLING project, we are making use of an Arabic-English bilingual lexicon to incorporate the LCS representations into an online Arabic dictionary.[12]

As we discussed above, the identification of "correct" meaning components is critical for the automatic acquisition of LCS representations. However, in order to arrive at these meaning components, we must ask the question: what belongs in a lexical representation for interlingual language tutoring and translation, and what does not belong in such a representation? The way we choose to represent words in the lexicon must be justified, both syntactically and semantically. This brings us to a related question: where is the dividing line between the lexical representation and the knowledge representation? Clearly, as was argued above (in the spirit of Levin), the lexical-semantic representation must be tied into the syntactic realization. Whenever we add more semantic features to our verbal classification, we are forced to think in terms of the syntactic ramifications of such an augmentation.

The work of (Wu and Palmer, 1994) has sought to examine such issues in the context of Chinese/English translation. In this work, the central concern is that of distinguishing among the many versions of *break* that arise in the Chinese language. In order to account for these distinctions, Wu and Palmer have added a dimension to Levin's classification that concerns the type of object separation that occurs during the breaking action. A difficulty with adding these components of meaning is that they have no independent justification in the syntactic structure of the verbs under consideration. An area of future investigation would be to augment the LCS acquisition program to include not only a mapping from syntactic behavior to components of meaning, but also from knowledge-based components of meaning to their LCS counterparts.

6.3 Verbs of Motion

In an attempt to better understand one of the larger classes of Levin's model, Verbs of Motion, we have investigated the approach of (Choi and Bowerman, 1991) in their analysis of both spontaneous and caused motion.

6.3.1 Compound Verbs Representing Spontaneous Motion

Choi and Bowerman note that in compound verbs representing spontaneous motion, the rightmost verb is usually 'kata'(go) or 'ota'(come). This verb is preceded by a Path verb, which in turn may be preceded by a Manner verb. (See table 1 on page 89.)

(33) John-i entek-ul (ttwi-e) oll-a ka-ss-ta -Nom mound-Acc run-Conn up-Conn go-Pst-Ind 'John went up the mound (running).'

Note that it is not the case that spontaneous motion is always represented in this way: Path and Motion (or Manner and Motion) can be represented by a single verb, as seen in the following:

(34) 'sanghaynghata' (go up)
'hahaynghata' (go down)
'ipcanghata' (enter)
'thoycanghata' (go out)
'kwanthonghata' (go through)....
'cilcwuhata' (run)
'phopokhata' (crawl)
'pihaynghata' (fly)....

---

[12] We are viewing this as a first approximation to a complete Arabic lexicon. We will be porting the same technique over to a Spanish lexicon in the next year.

6.3.2 Verbs Representing Caused Motion

Caused motion is expressed with (transitive) verbs that conflate [Motion + Path]. (See table 2 on page 91.) Choi and Bowerman conclude that Korean uses different lexicalization patterns for spontaneous motion and caused motion. We argue that this conclusion is questionable: Spontaneous motion can be represented by a single verb conflating [Motion + Path], as seen in (34) above. In addition, there are compound verbs which represent caused motion, as seen below:

(35) 'olly-e' (up) + 'ponayta'
'nayly-e' (down) + 'ponayta'
'tuly-e' (in) + 'ponayta'
'na-y' (out) + 'ponayta'
'nemky-e' (over) + 'ponayta'

6.4 Lexical Semantics and French Data

This section describes current collaboration with Dr. Patrick Saint-Dizier at IRIT in Toulouse. In this part of the project, we co-identified the primitives used in English definitions with those used in the French definitions. In so doing, we have defined a role-to-role mapping between each of the two languages.

6.4.1 Spatial predicates for Dorr's LCSs: English

Primitive	**Examples of Usage**
aboard	aboard
around	about, around
above	above, over
across	across
against	against
along	along, alongside
among	amid, among
apart	apart, away
around	around
at	at
away_from [at]	away, from
away_from at	away
back	back
back_toward [at]	back
behind	after, behind, in
between	between
beyond	beyond
co	along, with
down	down
east	east
from [at]	from
front	ahead, before, in
in	in
inside	inside, throughout, within
left	left
near	by, close, near
next	beside, next
north	north
not_co	without
off	off
on	on, upon
out	out
outside	outside
relative_to	in, related, relative, with
right	right
south	south
to [at]	to
to in	into
to off	off
to on	onto
toward [at]	to, toward, towards
toward at	for
toward in	into
toward on	onto
toward past	by, past
toward up	up
under	below, beneath, under, underneath
up	up
via [at]	by
via at	over, through, via
west	west
with	with

6.4.2 Spatial predicates for Saint-Dizier's LCSs: French

Primitive	**Examples of Usage**
aboard	a bord de
above	au dessus de, par dessus
across	au travers de
afar	loin de
against	contre
along	le long de
among	parmi
around	autour de, aux alentours, partout au travers de, partout autour de
at	a
behind	apres, derriere, en arriere de
between	entre
beyond	au dela de
co	avec
down	en bas
east	a l'est de
from at	a partir de, de
front	a l'avant de, au devant de, devant, sur le devant de
in	dans
inside	a l'interieur de, au dedans de, dans, partout dans
left	a gauche de
near	a proximite, aupres de, aux abords de, aux environs de, pres de
next	a cote de
north	au nord de
not_at	de
not_co	sans
on	sur
outside	au dehors de, en dehors de, hors de
relative_to	en ce qui concerne, en rapport avec, en relation avec, lie a, par rapport a, relativement a
right	a droite de
south	au sud de
to	a
to down	en dessous de
to in	dans
to up	jusqu a
toward at	en direction de, vers
toward back	en arriere, vers l arriere
toward in	dans
toward past	devant, vers
toward side	a cote de, non loin de, sur le cote
under	au dessous de, en dessous de, sous
up	en haut de
via	a travers
via at	dans, en passant par, par, via
west	a l'ouest de

7 Automatic Acquisition of Lexical Entries

One of the goals of this project is to construct a large-scale lexicon for Korean and English so that these can be used in PRINCITRAN. For many languages, dictionary resources are often scarce, possibly consisting of only a simple bilingual word-list. We describe techniques that are used to predict salient linguistic features of a non-English word by using the features of its English translation in a bilingual dictionary. While not exact, owing to inexact translations and language-to-language variations, these techniques can augment an existing dictionary with reasonable accuracy, thus saving significant time.

As noted by (Montemagni and Vanderwende, 1992), computational linguistics demands semantic information for parsing, yet little work has been done to extract and use such information on a large scale in a practical application. (See related remarks in (Wilks et al., 1989).) We demonstrate that it is possible to extract enough information from online resources for large-scale, automatic construction of semantic verb classes. The classes allow us to derive both lexical-semantic information (for composition of an interlingua) and thematic role information (for syntactic parsing as well as acquisition of new lexicons). While semantic classes are crucial for deriving the underlying meaning of a source-language sentence, thematic grids are a crucial component of syntactic analysis and lexical disambiguation.

Some have argued that the task of simplifying lexical entries on the basis of broad semantic class membership is complex and, perhaps, infeasible (see, e.g., (Boguraev and Briscoe, 1989)). However, there have been many arguments for the claim that certain aspects of syntactic behavior are predictable from meaning, e.g., thematic roles imposed on arguments (see (Fillmore, 1968), (Grimshaw, 1990), (Gruber, 1965), (Jackendoff, 1983), (Jackendoff, 1990), (Levin, 1993), (Pesetsky, 1982), (Pinker, 1989)). We take the position that it is at least feasible to predict thematic-grid information from combinations of syntactic behaviors, or "alternations" in the terminology of (Levin, 1993); and that some of these alternations can be derived from combinations of grammar codes in the Longman's Dictionary of Contemporary English (LDOCE) (Proctor, 1978).

While grammar codes have been used previously in automatic extraction tasks (see, e.g., (Wilks et al., 1990), (Boguraev and Briscoe, 1989)), the codes have been used primarily for the prediction of syntactic phrase structure, not for assigning thematic roles that are then used later for semantic analysis. Others that have used the LDOCE for automatic extraction (e.g., (Alshawi, 1989), (Wilks et al., 1989)) have focused on the derivation of semantic structures from definition analyses rather than the derivation of thematic-grid information from grammar codes. The use of bilingual dictionaries for the automatic construction of multi-language information from LDOCE is also currently under investigation by (Farwell et al., 1975). This work is closely related to ours in that it focuses on the extraction of broad semantic restrictions on arguments.[13] The work of (Sanfilippo and Poznanski, 1992) is even more closely related to our approach in that they attempt to create a large lexical database using semiautomatic techniques to recover syntactic and semantic information from machine-readable dictionaries. However, they claim that the semantic classification of verbs based on standard machine-readable dictionaries (e.g., the LDOCE) "a hopeless pursuit [since] standard dictionaries are simply not equipped to offer this kind of information with consistency and exhaustiveness." We take the view that it is possible to find a strong correlation between some LDOCE code combinations and semantic classes, and that the automatic realization of this correlation significantly aids the process of human verification of the resulting lexicon.

We have conducted two experiments. The first one was designed primarily to demonstrate the feasibility of building a database of thematic grids for over 6500 verbs. These verbs were taken from the English translations provided in a lexicon used in a related Army-funded project (for Arabic foreign language tutoring). Our approach in this experiment relies on the use of grammar codes from the LDOCE to derive the range of possible thematic-role assignments to verbal arguments.

The second experiment was designed to demonstrate that, using a combination of LDOCE codes and the verb classes of (Levin, 1993), it is possible to arrive at a more enriched semantic classification of at least 50% of these verbs and, hence, a more rich set of thematic grids. This experiment is based on the use of Levin's syntactic alternations in conjunction with the syntactic

---

[13] A hint is given (p. 143) that future investigation is headed in the direction of automatic extraction of "case roles", but no details are given.

codes in LDOCE, to uniquely determine semantic classes of verbs. While our initial attempts to map *single* subcategorization codes to thematic grids (in the first experiment) had very low yield, we found (in the second experiment) that we were able to find a correlation between *multiple* subcategorization codes with the Levin's semantic verb classes. In particular, the presence of certain code combinations allowed us to determine, automatically, the semantic classification of 72% of all Levin's verbs. We demonstrate that: (1) this yield is an absolute limit (upper bound) on accuracy for semantic classification of LDOCE verbs not occurring in (Levin, 1993); and (2) it is possible to approach this upper bound if we use linguistic techniques in conjunction with LDOCE codes to classify such verbs. In either case, human intervention will always be necessary for construction of a semantic classification from LDOCE.

The semantic verb classes resulting from this experiment map directly into a set of thematic grids that is richer than those produced by the first experiment. We expect that the hand verification of these enriched thematic grids would be at least as efficient as that of the first experiment because the improved grids provide the human checker (a native speaker of Korean) with much of the information that was missing from the "impoverished" grids. In either case, automatic construction of grids prior to hand checking is clearly more efficient than building the thematic grids from scratch.

The ultimate task is to make use of a Korean-English bilingual lexicon to incorporate the resulting thematic grids into the processing dictionary used in PRINCITRAN. We currently have a small bilingual lexicon from the Army Research Institute that contains 898 nouns, 299 verbs, 77 adverbs, 35 prepositions, and 15 adjectives. (See Appendix E.) We have incorporated thematic grids into this smaller one and will ultimately fill in thematic grids for and additional 6000+ words once we have expanded our base lexicon to a larger set. (A student will be hired next year specifically to enter bilingual data.)

The semantic verb classification that serves as the basis for this automatic acquisition exercise provides a number of benefits: (1) it takes minimal effort to add a new language as long as we have access to a bilingual dictionary containing English translations; (2) it enables automatic acquisition of lexical-semantic representations from which we can construct an interlingua; (3) it associates with each verb class a set of thematic grids that can be installed and used during syntactic analysis of the input sentence.

Described next is the first experiment, i.e., the automatic extraction techniques that we applied to the LDOCE in order to construct thematic grids for incorporation into a Korean-English dictionary. Section 7.2 describes the second experiment, i.e., our improvements on the automatic acquisition approach used in the first experiment. This experiment involved the use of verb classes from (Levin, 1993) in conjunction with the subcategorization codes in LDOCE.

7.1 Construction of Thematic Grids from LDOCE codes

Our current focus is on the enhancement of the sentence analysis component of PRINCITRAN. As with any sentence analyzer, the lexicon is a significant sub-component; thus, our primary objective is to construct a lexicon that is rich in both syntactic and semantic information so that we can provide a more enriched structural analysis as well as an adequate interlingua from which the target-language sentence will ultimately be generated. Our first experiment involved the mapping of a large set of English verbs (from (Levin, 1993)) into the corresponding thematic grids. Ultimately, these thematic grids will be stored in a 6000+ verb lexicon using automatic acquisition through bilingual entries.

7.1.1 Assignment of LDOCE-Based Thematic Grids to English Verbs

First, we describe the nature of the English verbs (from a corpus of 6500+) that serve as our input to the program for assigning thematic grids. Next, we present our technique of thematic grid compilation from the English dictionary.

We examined 6587 unique English translations from a lexicon used in a related Army-funded project (for Arabic foreign language tutoring). LDOCE contains extensive and useful information about argument structure for most of these verbs. We have used an on-line version of the LDOCE to extract verbs along with their syntactic "codes" (i.e., subcategorization frames). These were then used to produce a corresponding set of thematic grids for each verb entry in the lexicon.[14] We have mapped LDOCE codes into a set of thematic grids as shown in figure 10. By using the English translations in a Korean-English dictionary and the thematic grids derived from the LDOCE, we will ultimately install the resulting thematic grids automatically into a Korean verb database. (These will be verified by hand in the next phase of the project by a native Korean speaker.)

Out of the 6587 unique English entries, 2696 exactly matched a verb entry in the LDOCE. In these cases, our program directly incorporated the thematic grid from the code as given in figure 10. These were typically single verbs or single verbs followed by a preposition. Some examples are shown in figure 11.[15]

Of the remaining English entries, 2721 were what we call "parsed matches", i.e., these were matching phrases that contain a verb followed by a noun phrase, prepositional phrase, preposition or some combination thereof. Phrases that end with a preposition were considered transitive (T1), e.g. *abstain from*. Phrases that end with a noun phrase were considered intransitive (I), e.g. *adorn oneself*. Phrases that ended with prepositional phrases were handled according to the preposition: verbs associated with the prepositions *of*, *as*, or *by* were considered both transitive and intransitive (I,T1); verbs that were not associated with these prepositions were considered intransitive. Because *to* can be both a preposition and an infinitive marker, phrases that contained *to* were checked by hand. Some examples are shown in figure 12.

An additional 642 English entries were classified as "-ed past participles". These phrases consist of a verb followed by a '-ed' past particle. The past participles were considered to be adjectives and, thus, consumed one thematic role of the head verb. Generally this made the phrase intransitive (I), except in the case of *make*. Some examples are given in figure 13. There were 116 English entries that fell into the category of "-ly adverbs"; these consist of a verb followed by a '-ly' adverb. In such cases, the code of the head verb was used.[16] Some examples are given in figure 14. An additional 118 English entries were classified as '-s plurals'. These are phrases ending with plural nouns. The rules which apply to the "parsed matched" were applied here. Some examples are

---

[14] In addition to extracting verbs, other principle parts of speech were collected, including nouns, prepositions, adjectives and adverbs.

[15] In English, verbs that are both transitive and intransitive can be either 'ag ag_th' or 'th ag_th'. An example of 'ag ag_th' is *eat*, as in *John eats lunch*; here, the verb is 'ag_th', while in *John eats*, it is (ag). An example of 'th ag_th' is *freeze*; in *the water froze*, the verb is 'th', while in *John froze the water*, it is 'ag_th'. There is no reliable way to automatically distinguish these types of verbs, therefore, all intransitive thematic grids were assumed to be 'ag'. Note, however, that a more comprehensive theory of thematic roles *must* allow for alternative thematic grids corresponding to strict transitives and intransitives, especially if we are to consider languages other than Korean. For example, we are now considering the application of this technique to French data provided by Patrick Saint-Dizier. The second experiment addresses this concern by introducing an enriched set of thematic grids based on an extensive semantic classification.

[16] The presence of the adverb removes many of the possible verb readings in English because of the adjacency requirement for verbs and objects. For instance:

*John worked energetically Bill.

But adjacency is not a problem in the source language because these phrases represent a single verb. As such, the example given above might be rendered as:

John worked Bill energetically.

Consequently, many of the codes in the examples seem anomalous because of this adjacency difference.

Code	**Thematic Grid**	**Example**
I	ag	we PAUSED
I	th	the water froze
L9	ag_adv	she LIVES here
L1	ag_pred	she BECAME queen
L7	ag_pred	she BECAME famous
T1	ag_th	she KICKED the boy
X9	ag_th_goal	PUT it in the box
D1	ag_ben_th	GIVE the boy a book
X1	ag_th_pred	she CONSIDERED him her enemy
X7	ag_th_pred	she CONSIDERED him dead
I2	modal	I CAN fly
T2	ag_event+inf	I HELPED clean the window
I3	ag_prop-subj	he LIVED to be 90
L3	th_prop-subj	the difficulty IS to know what to do
T3	ag_prop-subj	I WANT to go
V2	ag_event+subj	he SAW her leave
V3	ag_prop+subj	he WANTS her to leave
I5	prop+that	it APPEARS that she will win
I6	prop+wh	it APPEARS as if she will win
L5	th_prop+that	trouble IS that you know bill
L6	th_prop+wh	it IS as if we had never met
T5	ag_prop+that	I KNOW that he will come
T6	ag_prop+wh	he DECIDED who should go
D5	ag_ben_prop+that	he WARNED her that he runs
D6	ag_ben_prop+wh	TELL me who is here
I4	ag_event+ing	she CAME running
L4	ag_event+ing	she ENDED UP dancing
T4	ag_event+ing	I ENJOYED singing
V4	ag_th_event+ing	he WATCHED her cooking dinner
L8	th_event+ed	he GOT trapped
I8	sj_event+ed	smoking IS not permitted
V8	ag_th_event+ed	he HAD a house built

Figure 10: LDOCE Codes, Thematic Grids, and Examples

English entry	**Code**	**Grid**
Anglicize	T1	ag_th
abandon	T1	ag_th
abate	I T1	ag ag_th
adhere to	T1	ag_th
aim at	T1 T4	ag_event+ing
band together	I	ag

Figure 11: English Entries that are "Exact Matches" with LDOCE Verbs

English Entry	**Code**	**Grid**
abstain from	T1	ag_th
act in unison with	T1	ag_th
adapt to	T1	ag_th
administer extreme unction to	T1	ag_th
accommodate each other	I	ag
achieve mutual understanding	I	ag
adorn oneself	I	ag
add water	I	ag
accuse of infidelity	I T1	ag ag_th
deprive of sleep	I T1	ag ag_th
appoint as candidate	I T1	ag ag_th
demand as a security	I T1	ag ag_th
divide by seven	I T1	ag ag_th
grab by the collar	I T1	ag ag_th
act with integrity	I	ag
be at a distance	I	ag
be in the middle	I	ag
pray for rain	I	ag
call to account	I T1	ag ag_th
call to prayer	I T1	ag ag_th
cause to decay	I T1	ag ag_th
come to a halt	I	ag
come to light	I	ag

Figure 12: English Entries that are "Parsed Matches" with LDOCE Verbs

English Entry	**Code**	**Grid**
be Anglicized	I	ag
be aroused	I	ag
become Africanized	I	ag
feel embarrassed	I	ag
get burned	I	ag
make bored	I T1	ag ag_th

Figure 13: English Entries Containing "-ed past participles" that Match LDOCE Verbs

English Entry	**Code**	**Grid**
act cautiously	I L1 L9 T1	ag ag_pred ag_adv ag_th
advance gradually	I T1	ag ag_th
approach gradually	I T1	ag ag_th
arrive constantly	I	ag
beat brutally	I L9 T1 X9	ag ag_adv ag_th ag_th_goal
behave indifferently	I L9 T1	ag ag_adv ag_th
close securely	I T1	ag ag_th
deal with harshly	D1 I T1	ag_th ag_ben_th
decrease gradually	I T1	ag ag_th
die prematurely	I L1 L7 L9 T1	ag ag_pred ag_adv ag_th
distribute equitably	T1	ag_th
do deliberately	D1 I I2 L7 L9 T1	ag_adv ag_th ag_ben_th ag_pred modal
render falsely	D1 T1 X7	ag_th ag_ben_th ag_th_pred
speak affectedly	I L9 T1	ag ag_adv ag_th
weave flimsily	I L9 T1 X9	ag ag_adv ag_th ag_th_goal
weigh heavily	L1 L9 T1 X9	ag_pred ag_adv ag_th ag_th_goal
work energetically	I L9 T1 X9	ag ag_adv ag_th ag_th_goal
wrap tightly	T1 X9	v ag_th ag_th_goal

Figure 14: English Entries Containing "-ly adverbs" that Match LDOCE Verbs

English Entry	**Code**	**Grid**
be an extremist	I	ag
be boring	I	ag
acclimate	I T1	ag ag_th
acquaint	I T1	ag ag_th
administrate	I T1	ag ag_th
adsorb	I T1	ag ag_th
dance the dabka	I	ag
deceive	I T1	ag ag_th
demobilize	I T1	ag ag_th
increase tenfold	I T1	ag ag_th
say "phew"	I	ag

Figure 15: English Entries not in LDOCE

Exact matches with LDOCE	2696
Exact parse (verb in LDOCE, followed by obj/pp)	2721
verb verb-ed (I, cf verb adj)	642
verb adj-ly (use arg of 'verb')	116
verb noun-s (I, cf verb obj)	118
not found in LDOCE (I/T1)	231
found, but not arg in LDOCE (I/T1)	63

Figure 16: Classification of 6587 English Entries

Korean	English	Thematic Grid
daejeobha	treat	I-WITH T1 WV5 X9
jareu	cut	ag_th
jara	grow	ag
jab	catch	ag ag_th ag_th_goal
jeulgi	enjoy	ag_event+ing
jina	pass	ag ag_th ag_th_goal
pal	sell	ag_ben_th
yeol	open	ag ag_th

Figure 17: Thematic Grids for Korean Verbs

appear in the heavens, *apply the brakes*, *be concerned with trifles*, *be covered with warts*, *flow in torrents*, *smack the lips*, and *use stratagems*. There were 231 remaining English entries with "missing" words. These were phrases containing words not found in the LDOCE. Frequently, these phrases contained spelling errors, derived words, foreign words, or unusual words. A small number were legitimate words that weren't available in the LDOCE. Verbs in this category were considered optional transitive/intransitive (I,T1).[17] Some examples are given in figure 15. Finally, there were 63 English entries found in LDOCE that did not have any thematic grid, i.e., there were no codes for these verbs. In these cases, the verbs were assigned the code (I,T1). Some examples are: *advert*, *almost*, *armor*, *attribute*, and *basil*.

We classified the 6587 English glosses in figure 16. We have recently started to incorporate these into the Korean dictionary. A small sample of the resulting entries for some of the Korean verbs in Appendix E is shown in figure 17.

Clearly, the thematic grids that have been acquired in this experiment provide minimal thematic relations based on simplistic subcategorization frames (e.g., transitive and intransitive). While subcategorization frames are useful for syntactic processing, they do not describe the full range of thematic possibilities that would be necessary for construction of an interlingua for machine translation. This was the motivation for moving toward more enriched thematic roles, as developed in the second experiment described in the next section.

7.2 Construction of Thematic Grids from Levin's Classification

While the information in the LDOCE has aided us significantly in our endeavor to construct a large-scale lexicon containing thematic grids, we are still left with the problem that many of these grids

---

[17] Two exceptions were words of the form '-ize', which were considered transitive and phrases of the form 'be ...',which were considered intransitive. The rules from the "parsed matches" were used in these cases.

Levin Verb	**Thematic Grid**
admire	exp_perc,purp
act	th_pred
believe	ag_th_prop
buy	ag_th,src,poss,ben
hold	ag_th,loc
keep	ag_th,loc
mix	th_goal ag_th_goal
pelt	ag_th,instr
pronounce	ag_th_pred
put	ag_th_goal
remember	exp_perc
remove	ag_th,src
tear	ag_th,manner

Figure 18: Thematic Grids Based on Levin's Verb Classification

Single-Word Verbs:	adjust, apprehend, approach, bear besiege, compel, deceive, distance, fail, harbor, immunize, lack, mutilate, ponder, reflect, secede, summon, sunbathe, venture, wrong
Multi-Word Verbs (be):	be a coward, be aware, be colorful, be delivered, be engrossed, be gathered, be in the sun, be lame, be newly acquired, be possible, be separated from, be suitable, be valid
Multi-Word Verbs (become):	become a Christian, become difficult, become numerous, become thin
Multi-Word Verbs (have):	have a bad character, have a cold, have no market, have pity, have weak eyesight

Figure 19: Random examples of Verb's not found in Levin

are highly language specific (e.g., "prop+that," which has no analog in Korean) and, moreover, that many thematic roles have not been included (e.g., source, location, time, etc.). We believe that a promising solution to this problem is to propose thematic grids on the basis of the verbal classification described by (Levin, 1993).

We adopt Levin's position that semantically related verbs have similar, or identical, argument structures (hence, thematic roles). We are currently using Levin's classification to build a set of thematic roles that are not only richer than the ones described above, but that are more language-independent. We have manually encoded the thematic grids for all 192 of Levin's semantic classes (3100 verbs, not including duplicates across the classes).[18] A small sample of these is shown in figure 18.[19]

Our intent is to "port" these thematic grids into a Korean dictionary using the same technique described above (based on English translations), thus replacing the LDOCE-based thematic grids with the richer version. Of the 6587 unique English entries used in the previous experiment, we

---

[18] Levin's classes are labeled with numbers ranging from 9 to 57. However, the actual number of semantic classes 192 (not 46) due to minor class subdivisions under each major class.

[19] The comma is used to indicate optionality of the subsequent roles.

found 3298 in (Levin, 1993); thus we have about 50% coverage of the English entries.[20] Of the 3289 missing words, it is interesting to note that 1710 English entries are of the form "be X," 161 are of the form "become X," 65 are of the form "have X," and 1353 are of some other form. As seen by these numbers, this is a potentially productive process. Random examples of verbs not found in (Levin, 1993) are shown in figure 19.

This experiment is intended to test the hypothesis that there exists a correspondence between combinations of syntactic codes in LDOCE and combinations of syntactic behaviors, or "alternations" in Levin's terminology. While our initial attempts to map *single* subcategorization codes to thematic grids (in the previous experiment) had very low yield, we found (in the current experiment) that we were able to achieve reasonable results by mapping *multiple* subcategorization codes into Levin's semantic classes.

We show that the presence of certain code combinations allowed us to determine, automatically, the semantic classification of 72% of Levin's verbs. We then demonstrate that: (1) this yield is an absolute limit (upper bound) on accuracy for semantic classification of LDOCE verbs not occurring in (Levin, 1993); and (2) it is possible to approach this upper bound if we use linguistic techniques in conjunction with LDOCE codes to classify such verbs. In either case, human intervention will always be necessary for construction of a semantic classification from LDOCE.

7.2.1 Assignment of Levin-Based Thematic Grids to Lexical Entries

Our first task in this experiment was to determine an upper bound on the ability to classify LDOCE verbs not occurring in (Levin, 1993). This experiment used a brute-force (hence non-robust) technique for semantic classification based on a correlation between unique LDOCE code combinations and Levin's semantic classes.

We started by examining LDOCE-coded verbs that *do* occur in (Levin, 1993). For each such verb, we determined the probability that the verb (with its associated LDOCE codes) belonged to a given Levin-based class. We called this probability $L(\text{class}|\text{codes})$, i.e., the probability of predicting a Levin-based class given a list of LDOCE codes, where the Levin-based class with the highest probability is the best choice, the next highest probability is the next best choice, etc. The objective was to set $L(\text{class}|\text{codes})$ for every class-codes combination so that the resulting classification would be as close as possible to the "real" classification (i.e., the classification provided by Levin). No regard was given to anything else other than achieving a maximally correct verb classification. By setting $L(\text{class}|\text{codes})$ so that an optimal classification would be achieved, we were able to determine the upper bound on the ability to use LDOCE codes to classify verbs *not* occurring in Levin's classes. (We will show this below.)

The optimal value for each $L(\text{class}|\text{codes})$ was defined on the basis of a frequency count on occurrences of LDOCE-based code patterns (abbreviated "codes") in the Levin-based classes (abbreviated "class"). We first constructed a list of all unique LDOCE-based code patterns and then determined the number of times each pattern appeared on entries in each Levin-based semantic class. There were 925 unique LDOCE-based code patterns, over half (464) of which occur only once, and 849 of which occur 5 times or less. There were 2775 words representing 3789 entries in Levin's classes (some words occur in multiple classes).

For a given semantic class and code pattern, we defined $L(\text{class}|\text{codes})$ to be equal to the number of times that code pattern appeared in that class, divided by the number of times that code pattern appeared in all classes. Figure 20 shows a small sample of the 925 unique LDOCE code patterns (under the column labeled **Pattern**) and their associated top 5 Levin-based semantic classes (under the column labeled **Levin Classes**). The column labeled **All** corresponds to the total number of occurrences of a LDOCE-based code pattern in the Levin-based classification; the column labeled **Top 5** corresponds to the total number of occurrences in the top 5 semantic classes.

---

[20] This coverage includes 1791 single-word entries and 1507 multi-word entries.

Pattern	Levin Classes	All	Top 5
I T1 N	21 (37.3) 16 (45.4) 12 (43.2) 11 (38) 9 (22.4)	265	69
T1 N	31 (9.9) 20 (31.1) 15 (22.4) 13 (9.8) 13 (9.10)	246	92
T1	52 (31.1) 16 (33) 14 (45.4) 10 (44) 9 (10.5)	220	101
I T1	85 (45.4) 6 (47.2) 5 (43.2) 5 (48.1.1) 4 (10.5)	188	105
N	20 (9.9) 15 (51.4.1) 12 (13.7) 9 (10.7) 9 (51.5)	145	65
I N	18 (38) 12 (37.3) 9 (36.1) 9 (43.2) 8 (40.2)	120	56
I	5 (51.3.2) 4 (48.1.1) 3 (45.4) 3 (45.5) 3 (48.2)	56	18
I T1 X9 N	4 (22.4) 3 (10.4.1) 3 (11.4) 2 (10.7) 2 (17.1)	52	14
I L9 T1 X9 N	5 (51.3.2) 3 (22.4) 3 (47.3) 2 (17.1) 2 (18.4)	47	15
L9 T1 N	5 (51.3.2) 3 (43.4) 2 (9.9) 2 (35.5) 1 (9.5)	33	13
L9 T1 X9 N	3 (9.3) 2 (10.4.1) 2 (17.1) 2 (18.1) 2 (22.4)	32	11
L9 N	9 (51.3.2) 4 (56) 2 (30.3) 2 (43.2) 2 (47.6)	28	19
L9	13 (51.3.2) 4 (47.1) 2 (46) 1 (31.3) 1 (36.1)	27	21
T1 T1-WITH N	7 (9.9) 6 (9.8) 2 (31.1) 2 (47.8) 1 (9.3)	23	18
T1 WV4	17 (31.1) 2 (39.4) 1 (27) 1 (33) 1 (47.8)	22	22
T1 T1-FROM	7 (10.1) 3 (10.5) 2 (10.4.1) 2 (10.6) 2 (23.1)	20	16
I WV4 N	2 (40.2) 2 (43.1) 2 (47.2) 2 (47.6) 1 (31.3)	19	9
X9 N	4 (22.4) 2 (29.8) 1 (9.1) 1 (9.7) 1 (10.4.1)	17	9
L9 X9 N	2 (51.3.2) 1 (9.2) 1 (11.5) 1 (19) 1 (22.3)	16	6
I T1 WV4 N	3 (31.1) 2 (43.2) 2 (47.8) 1 (20) 1 (21.2)	15	9
I I-WITH T1 N	5 (43.2) 2 (36.1) 1 (37.3) 1 (38) 1 (40.2)	14	10
I L9 T1	2 (45.4) 1 (9.5) 1 (9.7) 1 (22.3) 1 (26.1)	13	6
I I-ABOUT N	2 (13.7) 2 (37.3) 2 (37.6) 2 (38) 1 (37.8)	12	9
I I-FOR T1 N	2 (35.1) 2 (35.2) 1 (10.9) 1 (13.7) 1 (31.3)	11	7
T1 T1-OF	9 (10.6) 1 (45.4) 0 () 0 () 0 ()	10	10
D1 D1-FOR I L9 T1 X7 X9 N	1 (9.6) 1 (11.2) 1 (22.3) 1 (23.2) 1 (26.1)	9	5
I T1 T1-WITH N	2 (22.2) 1 (9.7) 1 (9.8) 1 (36.1) 1 (43.4)	8	6
I T1 T5 N	3 (37.3) 1 (37.7) 1 (38) 1 (43.2) 1 (55.1)	7	7
D1 D1-FOR D1-TO I I-FOR T1 T1-TO T1-WITH T4 X1 X7 X9 N	1 (13.3) 1 (13.4.1) 1 (13.5.1) 1 (15.2) 1 (51.1)	6	5
D1 D1-FOR D1-TO I L1 L7 T1 T1-TO V3 X7 X9 N	1 (10.5) 1 (11.3) 1 (29.2) 1 (54.2) 1 (54.3)	5	5
I L9 T1 T1-ON WV5 X9 N	1 (10.1) 1 (12) 1 (23.2) 1 (25.2) 0 ()	4	4
I L9 T1 T1-WITH N	1 (17.2) 1 (51.3.2) 1 (57) 0 () 0 ()	3	3
D1-ON D1-OVER D1-WITH I T1 N	2 (9.7) 0 () 0 () 0 () 0 ()	2	2
T1 WV5 WV5-WITH N	1 (9.9) 0 () 0 () 0 () 0 ()	1	1
T1 T1-IN T1-WITH X7 N	1 (9.9) 0 () 0 () 0 () 0 ()	1	1
T1 T1-IN T1-WITH WV5 N	1 (9.9) 0 () 0 () 0 () 0 ()	1	1

Figure 20: Sample of Unique LDOCE Code Patterns and with Top 5 Levin-based Semantic Classes

Choice					**Total**	**Total**
1st	2nd	3rd	4th	5th	(out of 3789 entries)	(out of 2775 words)
1402	587	331	200	139	2659 (70.2%)	2009 (72.4%)

Figure 21: Semantic Classification Based on Brute-Force Strategy

The probability $L(\text{class}|\text{codes})$ is defined to provide values that are guaranteed to give us a classification that is optimal. To show this, we consider an example. As shown in figure 20, the LDOCE-based code pattern {I T1 N} appeared 265 times. This pattern occurred 21 times in the semantic class 37.3, the most common class for this code pattern. Therefore, $L(\text{class} = 37.3|\text{codes} = \{\text{I T1 N}\}) = 21/265$.

When we attempt to classify the pattern {I T1 N}, $L(\text{class}|\{\text{I T1 N}\})$ will equal the number of times {I T1 N} occurred divided by 265. The class with highest probability is simply the class with the highest occurrence, (eg "37.3" in the example above). In this way we have correctly classified the maximum number of verbs. Put another way: if any other class were selected as the best first guess, fewer verbs would have been correctly classified. Likewise, the best second choice is also chosen, etc.

7.2.2 Results of Brute-Force Classification Strategy

We tested the classification strategy given above on the 3789 English entries that occur in Levin's classes. (The intent, then, is to compare the results with the actual classification given in (Levin, 1993).) The number of entries correctly classified is shown in figure 21. Of the 3789 entries, 2659 (70.2%) were correctly classified in one of the 5 best classes. Of the 2775 words, 2009 (72.4%) had at least one class correctly chosen.

The reason this result is of interest is that, while the method is unprincipled (i.e., it is a brute-force technique), it does serve to show that the upper bound for *any* classification scheme based solely on LDOCE codes is ~72%. That is, we will never do better than this, no matter how linguistically principled we are in our interpretation of LDOCE codes.

If we were to achieve 72% accuracy consistently from our automatic semantic classification approach, our efforts to build an Korean dictionary would receive a considerable boost in that the human checker would only have to make changes less than 30% of the time. However, we must consider whether the method described above is is useful for the acquisition of novel verbs, i.e., those *not* occurring in (Levin, 1993). Clearly, a flaw of this approach to classification is the lack of robustness. In particular, as mentioned above, over half (464) of the 925 unique LDOCE patterns occurred only once. One might wonder how many additional unique LDOCE patterns exist (outside of the ones corresponding to Levin's verbs) and what we should do when such patterns are encountered. The current classification scheme does not handle new LDOCE patterns. An analysis of the remaining verbs in the LDOCE indicates, in fact, that there are over 600 additional unique codes (a total of 1524 unique codes). Thus, the ability to handle new codes is clearly an important issue that we must address. (We will return to this point in the next subsection.)

An additional flaw of this approach is that, even if the number of new LDOCE patterns were low, the fact that half of the 925 unique LDOCE patterns occur only once indicates that the 72% statistic is spuriously high; obviously, it is not difficult to guess the semantic class of a verb whose LDOCE code occurs only once — the class is, in fact, always correctly determined in these cases. This is an inherent danger of *any* analysis that assumes a predictable input (in this case, a perfect code-to-class correspondence even for verbs whose code is unique). When we consider the assumptions we made to obtain the above results, it is even easier to believe (intuitively) that the 72% statistic is the best we will ever be able to do with LDOCE codes.

The question to consider now is why we cannot achieve a higher upper bound on the semantic classification process. One limitation of our method that is within our grasp (and that will be addressed in future versions) is the occurrence of multiple word meanings. For example, the verb *sleep* appears in Levin's classification as a "Snooze" verb (as in *Gloria slept*) as well as a "Fit" verb (as in *The cabin sleeps 5 people*). Both of these senses of *sleep* appear in the LDOCE (the first with the code I and the second with the code T1), yet our current algorithm collapses these two into the same code pattern {I T1}.

We speculate that there are (at least) two difficulties beyond our control, which make a higher statistic unattainable from the LDOCE-Levin coupling. The first is simply that there are errors and omissions in the LDOCE. For example, the verbs *grease* and *crown* are in the same verb class in (Levin, 1993), yet only one of them, *crown*, contains the code WV5, which indicates that the "-ed" form can be used as an adjective. (Clearly, the verb *grease* can also be used as an adjective, yet this information is not indicated in LDOCE.) There is no single LDOCE code pattern corresponding to these two verbs; thus, an erroneous semantic classification is possible for at least one of them.

The second difficulty is that there is an inherent mismatch between the syntactic tests that Levin considers in order to distinguish among her semantic classes and the syntactic subcategorization information available in the LDOCE. In fact, we implicitly arrived at this conclusion already upon completion of the first experiment. That is, we determined that certain LDOCE codes (e.g., those corresponding to "transitive" and "intransitive") simply are not sufficient for the generation of enriched (e.g., Levin-based) thematic grids. This implies that the LDOCE codes do not provide enough information to cover all of the combinations of syntactic behaviors required for distinguishing between the semantic classes in (Levin, 1993). For example, two crucial syntactic tests that Levin uses for distinguishing semantic classes are the Unspecified Object Alternation (e.g., that it is possible to say *John ate* as well as *John ate the apple*) and the Causative alternation (e.g., that it is possible to say *John broke the vase* as well as *The vase broke*). However, these two tests are collapsed into a single LDOCE code pattern: {I T1}. Thus, as far as the LDOCE is concerned, these occur in the same semantic class.

In the next section, we address the concern (mentioned above) that our brute-force techniques are not robust enough to handle new LDOCE code patterns.

7.2.3 Application of Linguistic Techniques for Levin-Based Classification

As we have already seen, we will never be able to surpass 72% accuracy using *any* semantic classification technique based on LDOCE codes.[21] However, it might be possible to develop a more robust technique that approaches this statistic *and* that is applicable to new LDOCE code patterns as they are encountered. Thus, the second task of the current experiment was to apply a more linguistically motivated approach to verb classification, in an effort to broaden the applicability of the acquisition method. With this approach we were able to achieve 55% accuracy of semantic classification (automatically checked against the verbs occurring in (Levin, 1993)).

The approach is to analyze LDOCE codes not in terms of a correspondence between multiple code patterns and semantic classes, but as a correspondence between individual codes and semantic classes. We believe this to be the first step toward the application of linguistic techniques for building a Levin-based semantic classification. In particular, the results of this experiment have revealed a correlation between syntactic alternations and LDOCE codes that might be exploited in future experiments.

We first built a cross reference of Levin's classes with LDOCE codes. Next we determined what codes were applicable to which classes and developed a correlation function (CORR) that matches a code to a class. The probability that a code occurs on any given entry is the total number

---

[21] This ignores the fact that sense disambiguation (e.g., distinguishing between two types of *sleep* verbs given above) might bring up this number. Whether there is a significant gain achieved by taking differing word senses into account remains to be seen in future experimentation.

of occurrences of that code, divided by the total number of words in the system. Therefore, the number of occurrences we would expect for a particular code in a particular class is the probability of the code times the number of entries in the class.

The correlation function is based on the number of entries a particular code occurs with and the number of entries we would expect it to occur with if it were randomly distributed. Intuitively, if an entry occurs with a code the number of times we would expect, given the size of a particular semantic class, then the entry is uncorrelated with respect to that class. If the entry occurs with a code occurs more than expected, it is positively correlated. The precise CORR function is actually the probability that a code would occur fewer times than actual, minus the probability that it would occur more times than actual:

$$\begin{aligned}&\mathrm{CORR}(\mathrm{code},\mathrm{class}) = \\ &\quad P(n < \mathrm{Count}(\mathrm{code},\mathrm{class})|\mathrm{code},\mathrm{class}) - P(n \geq \mathrm{Count}(\mathrm{code},\mathrm{class})|\mathrm{code},\mathrm{class})\end{aligned}$$

The values of CORR range over [-1.00,1.00]. A bimodal distribution is assumed in the calculation of $P(n|\mathrm{code},\mathrm{class})$.

As an example, consider the class 9.1 in (Levin, 1993), i.e., Verbs of *Putting*. This class has 12 entries. (Entries that do not occur in LDOCE are not considered.) 7 of these entries contain the X9 code. If X9 had been randomly distributed to this class, we would expect to find 1 occurrence of it. The CORR for X9 in the 9.1 class is 1.00, a very strong indicator that this code is applicable to this class. On the other hand, N occurs on 7 entries in class 9.1. But we would expect it to occur 8 times if it were randomly distributed to this class. Consequently, the CORR for N is low, -0.31; it is essentially uncorrelated with this class.

The method described above allows strong correlations to be posited even if only a small number of entries in a class contain a given code. For instance, the semantic class 9.7 (i.e., *Spray/Load* verbs) has 44 entries, 11 of which occur with the LDOCE code T1-WITH. Even though only $\frac{1}{4}$ of the entries have this code, the code has a CORR value of 1.00.

We used the CORR function as part of a experiment that relied on Bayes equation to predict a semantic class given a set of LDOCE codes:

$$M(\mathrm{class}|\mathrm{codes}) = \frac{P(\mathrm{codes}|\mathrm{class})P(\mathrm{class})}{P(\mathrm{codes})}$$

Given a set of codes, we can calculate $M(\mathrm{class}|\mathrm{codes})$ for every class and pick the class that has the highest probability (M). In order to do this we must be able to calculate the terms on the right side, $P(\mathrm{codes}|\mathrm{class})$, $P(\mathrm{class})$, and $P(\mathrm{codes})$. $P(\mathrm{class})$ is the number of entries in the class divided by the total number of entries. $P(\mathrm{codes})$ is the product of the independent probabilities of each code. $P(\mathrm{codes}|\mathrm{class})$ is the probability of a particular set of codes being produced from a member of the specified class. In order to calculate $P(\mathrm{codes}|\mathrm{class})$ we must be able to assign a value to $P(\mathrm{code}|\mathrm{class})$, the probability of a single given code appearing on an random entry in class. This probability depends on whether the code is correlated with the class, i.e., it is determined by the CORR function described earlier.

The CORR function allows us to introduce the notions of *marked* and *unmarked* classes. A class is *marked* if the CORR(code,class) is greater than some threshold (e.g., 0.8); it is *unmarked* otherwise. Once we have marked all the classes, we can determine the probability of a code appearing on an entry in a marked or unmarked class. If the class is marked for the code, then the probability of the code occurring is the number of times it occurred on entries in marked classes divided by the total number of entries in marked classes. If it is unmarked then the probability is the number of times it occurred on entries in unmarked classes divided by the total number of entries in unmarked classes.

To clarify this technique, we consider an example. There were 3828 total entries. The code D1 occurred 217 times in the entire set. Of those, 162 occurrences were on entries in classes that were marked for D1. There were a total of 556 entries in the classes that were marked for D1.

Therefore, the probability of D1 occurring on a random entry is $\frac{217}{3828}$ (0.056688). The probability of it occurring on an entry in a class that is marked for D1 is $\frac{162}{556}$. The probability of D1 occurring on a class unmarked for D1 is $\frac{(217-162)}{(3828-556)}$ (0.016809). D1 is 17 times more likely to occur on an entry in a class that is marked for D1 that it is to occur elsewhere.

Using the above formulation of $M(\text{class}|\text{codes})$, we attempted to classify verbs. We calculated $M(\text{class}|\text{codes})$ for every class and recorded the 5 best scores. The raw results of this experiment are shown in figure 22. Correctly predicted classes are marked with an asterisk (*). Figure 23 shows a comparison of the results of this experiment with the actual classification given in (Levin, 1993) Of the 3789 entries, 1807 (∼47%) were correctly classified in one of the 5 best classes. Of the 2775 words, 1527 (55.0%) had at least one class correctly chosen. The results of this experiment have also revealed a correlation between syntactic alternations and LDOCE codes that might be exploited in future experiments. For instance, the LDOCE code N (noun) occurs in classes that have zero-related Nominals, which is a key test for class membership in (Levin, 1993).

The advantage of this approach over the brute-force technique is that it is robust; any new code pattern that is encountered will be classified on the basis of a correlation between syntactic information and Levin's semantic classes. It is possible that some combination of the brute-force technique and the linguistically motivated approach will provide an optimal, robust system for lexical acquisition of verb-class information. We expect that more experimentation will be required in order to determine the threshold for application of one technique over another.

The final phase of this experiment involves the incorporation of the enriched version of the thematic grids derived from Levin's semantic classes (e.g, see figure 18) into the Korean lexicon. As before, our intent is to install these automatically (on the basis of English translations) and to check the resulting entries by hand. This portion of the experiment will be conducted once the method of semantic classification has stabilized (i.e., after we have experimented with thresholds for a combination of brute-force and linguistically motivated techniques).

One of the benefits of using the Levin-based semantic classification is that, just as we are able to derive a set of thematic grids (as shown in figure 18), we are also able to associate a lexical-semantic representation with each verb that is semantically classified. These representations serve as input to a composition process that derives the interlingua (based on earlier work by (Dorr, 1993b)) for the translation process. One of the areas for future work is the augmentation of the Korean lexicon (using the English translations) so that lexical-semantic representations are included with verbal entries. This will be achieved by filling out lexical-semantic templates associated with each of Levin's semantic classes and installing the resulting forms in the corresponding lexical entries for Korean.

An area of future research is an analysis of LDOCE codes in terms of code *pairs* rather than in terms of individual codes. We expect that examining code pairs would allow us to discover implicit encodings of Levin's alternations, e.g., the pair {I T1} is frequently used in cases where the Causative alternation is available. Note, however, that as the number of codes increases, the robustness of the method decreases due to the potential for encountering unknown code patterns. Determining where this tradeoff becomes significant is one of the areas we will be investigating.

Another area to address in future versions of the lexical acquisition procedure is the existence of polysemy in the English translations provided in the Korean lexicon. Polysemy is a problem because it leads to a many-to-one mapping in both directions between target language items and their English translations (i.e., many target senses for one English word or one target sense for many English translations). If one sense maps to many senses in the target language, then the associated LDOCE codes will end up with the wrong semantic mapping and the word will be misclassed. There is no way to forestall using the LDOCE codes; thus, we must experiment with post hoc checking techniques (providing the human checker with a smaller number of post hoc tests to separate out incorrectly classed items). The tests would be based on a subset of diagnostic alternations from Levin's book to aid in correction or augmenting codes with selectional or other

Word	**LDOCE Codes**	**Predicted Levin Classes**
ABANDON	T1 T1-TO N	22.4 29.7 42.1 22.3.c 13.2
ACCLAIM	T1 T1-AS X1 N	33.a* 25.3 29.2 37.7 29.1
BIKE	I N	51.3.2 9.9 38 37.3 43.2
BLEED	I I-FOR T1 T1-FOR	40.1.2* 41.1.1 31.3.c* 33.b 37.7
CAROL	I T1 N	31.1 45.4.d 51.3.2 9.9 38
CARPET	T1 N	31.1 51.3.2 9.9* 9.8* 22.4
CLUTTER	T1 N	31.1 51.3.2 9.9 9.8* 22.4
CONCEAL	T1 T4	33.b 31.2.b * 22.3.c 52
DEEM	X1 X1-TO-BE X7	37.7 29.3 9.5 37.6 29.2
DEPOPULATE	T1 WV5	31.1 9.8 5.3 45.4.f 47.8
ENRICH	T1 T1-BY T1-WITH	9.8* 47.8 1.1 23.1 33.b
EXPIRE	I	45.4.g 49 6.1 45.5 43.4
FREE	T1 ADJ	47.8 31.1 2.2.a 13.5.2 45.4.c
FUNNEL	L9 T1 X9 N	9.7 51.3.2 0.4.1 18.1 9.3*
HANG	I L9 T1 N	51.3.2 43.2 3.4 30.3 47.2
HEEL	I T1 N	31.1 45.4.d 51.3.2 9.9* 38
HOOK	I T1 X9 N	22.4* 18.2 31.1 47.6 45.4.d
ILLUMINATE	T1 T1-WITH WV5	9.8 47.8 9.9 31.1 25.3*
KICK	I T1 X9 N	22.4 18.2 31.1 47.6 45.4.d
KIDNAP	T1	31.1 9.8 10.5* 10.6 33.b
LADLE	T1 T1-INTO T1-OUT-OF N	10.6 9.7 21.2 31.1 9.3*
LESSEN	I T1	31.1 45.4.d* 45.4.a 45.4.b 9.8
MEASURE	I L1 T1 N	30.3 26.4 47.5.2 22.2.a 47.6
MOAN	I I-ABOUT T1 T5 N	38 37.3* 37.8 31.3.b 37.6
NETWORK	N	51.3.2 9.9 10.7 9.10 51.4.1
NUMB	T1 WV4 WV5 WV5-WITH ADJ	31.1* 45.4.a 9.9 10.5 51.3.2
PANEL	T1 T1-IN T1-WITH WV5 N	9.9* 9.8 47.8 22.2.a 25.1
PANT	I I-AFTER I-FOR I3 T1 N	40.1.1* 35.6
PEEL	I T1 X9 N	22.4 18.2 31.1 47.6 45.4.d
PRAWN	I N	51.3.2 9.9 38 37.3 43.2
PULVERIZE	I T1	31.1 45.4.d 45.4.a 45.4.b 9.8
RECREATE	T1	31.1 9.8 10.5 10.6 33.b
REVERBERATE	I	45.4.g 49 36.1 45.5 43.4
SHOUT	I T1 T5 N	38 37.3* 43.2 37.8 35.2
SLUMBER	I N	51.3.2 9.9 38 37.3 43.2
SWING	I I-FOR L7 L9 T1 X7 X9 N	51.3.1.a* 47.7 47.6* 50 26.5
TOTE	T1 N	31.1 51.3.2 9.9 9.8 22.4
UNZIP	T1	31.1 9.8 10.5 10.6 33.b
VOMIT	I T1 N	31.1 45.4.d 51.3.2 9.9 38
WORSEN	I T1	31.1 45.4.d* 45.4.a 45.4.b 9.8

Figure 22: Sample Verb Classification Based on Linguistic Correlations

Choice					**Total**	**Total**
1st	2nd	3rd	4th	5th	(out of 3789 entries)	(out of 2775 words)
814	379	262	196	156	1807 (46.6%)	1527 (55.0%)

Figure 23: Semantic Classification Based on Linguistic Correlations

disambiguating features before checking.

8 Interface Between LCS in KR

Another problem that we have investigated for this project is the use of a concept-based approach to lexical selection, i.e., the task of choosing an appropriate target-language term in generating text from an underlying meaning representation. This investigation has aided us in our quest to determine precisely where the LCS (interlingua) interfaces with a full-blown knowledge representation scheme.

Our goal is to provide a linguistically motivated scheme for handling lexical mismatches; we view this not solely as a language-to-language problem in the domain of machine translation, but as the larger problem of handling "lexicalization gaps" in the mapping from the knowledge base to the surface realization.

We describe an architecture for lexical selection in interlingua-based machine translation (MT) systems using KL-ONE-like concepts (Brachman and Schmolze, 1985) for grounding the lexical semantic descriptions of both target-language (TL) and source-language (SL) words. The concepts are classified into a semantic ontology for each supported language using LOOM, a KL-ONE-like term classifier. We believe this architecture is an advance over previous designs for three reasons: (*i*) It provides a more precise method for identifying and resolving mismatches between SL and TL words; (*ii*) It facilitates a graceful combination of the knowledge base (KB) and the lexical-semantic representation; and (*iii*) It offers greatly increased semantic expressive power, enabling the representation of complex constraints and relationships between intra-concept constituents.[22]

With respect to representing word meaning, two camps in MT research have prevailed: (*i*) lexical conceptual structures (LCS), in the spirit of (Jackendoff, 1983); and (*ii*) frame-based systems in the spirit of (Nirenburg et al., 1992). We combine the most promising aspects of both approaches, i.e., the minimalism and decompositionality of LCSs,[23] as well as the ability to perform frame-based reasoning in an object-oriented environment. We take the view that, as semantic lexicographers, we can benefit from the techniques used by both LCS and knowledge representation (KR) experts in designing semantic representations of word meaning. Specifically, we use LOOM (MacGregor, 1991), a KL-ONE-like term classifier, and its concept definitions — reasoning frames with the added power of logical constraints.[24]

Our approach is interlingual (i.e., language independent); our goal is to demonstrate the feasibility of encoding a lexical-semantic representation by means of concepts in a KB, thus allowing the interlingua (IL) to capture aspects of both types of information. We feel that the proposed architecture not only meets this original goal, but is a decisive foundation for increasing the expressive power of the lexical semantic representation and hence the precision of the surface realization. One important result is the ability to deal precisely with the case where the target language lacks a word that directly matches the IL fragment to be generated in the target language.

---

[22] Intra-concept constituents are stored within the concept roles and are constrained by well-formedness conditions.

[23] We use the term "decompositionality" to mean the representation of concepts as combinations of atomic units of meaning.

[24] Barnett, Mani, and Rich (Barnett et al., 1994, p. 363) suggest a similar use of a term classifier.

We should point out that although the focus of this technique is on machine translation (MT), it would work as well for knowledge-based language generation.[25]

8.1 A Brief Architectural Description

Like (Nirenburg et al., 1992), we seek to determine a range of semantic relations among words[26] in our semantic ontology. Our approach differs, however, in that it uses a term classifier to set up an ontology of lexical entries for each supported language. For a given target language, its ontology helps to determine, through classification, the lexical entry which most closely matches the portion of the IL form being generated.

Before translation runtime, the classifier is employed to form separate lexical entry ontologies for each supported language. At translation runtime, the lexical selection process attempts to realize a given portion of an IL (CLCS)[27] by running the term classifier on the relevant IL structure and then determining where the concept falls in the ontological hierarchy.[28]

Consider an ultra-minimal semantic ontology, shown in Figure 24, that consists of seven lexical-semantic forms (RLCS's) and their respective lexicalizations in English and German. (We have placed '***' in the figure where no word for the particular language exactly matches the forms listed.) The German words are *veranlassen, bewegen, transportieren,* and *fahren* and the English words are *cause, move, transport,* and *bus*. Prior to translation runtime, the word descriptions are classified and the ontology in Figure 24 is produced.

We now consider the runtime process of lexical selection during the translation of the German word *bewegen* into the English word *move.* First, the RLCS for *bewegen* is retrieved and composed with other sentence elements to form the CLCS in the analysis phase of the translation. Then in the generation phase as part of the algorithm for selecting a TL word, the same term classifier that was used to build the English lexicon is re-used to determine where the relevant portion of the CLCS falls in the ontology. Since *bewegen* is synonymous with one sense of the word *move*, the classifier will find that this portion of the CLCS matches the ontology entry for the causative sense of the word *move;* this will then be realized in English as *move*.

This example has been kept simple for expository reasons, but section 8.4 outlines the more complicated cases, i.e. where the matching fails.

8.2 Lexical Mismatches

We now discuss cases of mismatch between the source and target languages. The mismatch problem in MT has received increasingly greater attention in recent literature (see (Dorr, 1994), (Dorr and Voss, 1993), (Barnett et al., 1994), (Beaven, 1992), (Kameyama et al., 1991), (Kinoshita et al., 1992), (Lindop and Tsujii, 1991), and (Whitelock, 1992) as well as related discussion in (Melby, 1986) and (Nirenburg and Nirenburg, 1988).) In particular, (Barnett et al., 1994) divide distinctions between the source language and the target language into two categories: translation *divergences*, in which the same information is conveyed in the source and target texts, but the structures of the sentences are different (as in previous work by (Dorr, 1994)); and translation *mismatches*, in which

---

[25] See (Stede, 1993) for further discussion with respect to multilingual generation from a knowledge base.

[26] Strictly speaking, the lexicons in interlingua-based MT systems are not restricted to word-level entries. For the purposes of this paper however we will refer to "words" in the lexicons, setting aside the details about other types of lexical entries. See (Levin and Nirenburg, 1993) for further discussion on extending the range of lexical entries in MT systems.

[27] Composed Lexical Conceptual Structure or instantiated Root Lexical Conceptual Structure (RLCS). RLCS's are the actual semantic concept descriptions for the words in a lexicon. The terms RLCS and CLCS are borrowed from (Dorr, 1993b), (Dorr, 1994).

[28] In this paper we focus on the lexical selection component of the generation process. That is, we address the problem of selecting the appropriate TL word after the CLCS has been partitioned into relevant components for structural realization.

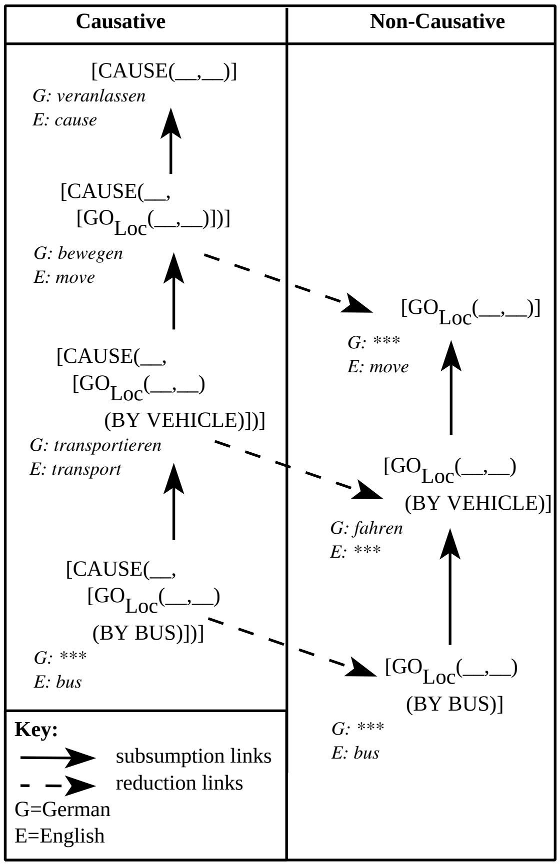


Figure 24: RLCS's with Directed Subsumption Links and Crossover Reduction Links in Semantic Ontology

the information that is conveyed is different in the source and target languages (as in (Kameyama et al., 1991)).

Both types of distinctions must be addressed in translation, yet most MT researchers have ignored one or the other. Researchers investigating *divergences* (see, e.g., (Dorr and Voss, 1993)) are more inclined to address the mechanism that links the IL representation to the syntactic structure of the target language, whereas investigators of the *mismatch* problem (see, e.g., (Barnett et al., 1994),(Kameyama et al., 1991)) are more inclined to focus on the details of the conceptual representation underlying the IL. The novelty of our approach is that it addresses the problem of mismatches through access to the KR while retaining enough of the structure of the IL to resolve the divergence problem. We focus on the problem of mismatches for the remainder of this paper; the reader is referred to (Dorr, 1994) for an in-depth treatment of the divergence problem.

Our solution to the mismatch problem involves the use of the LOOM KL-ONE-like term classifier. The approach is similar to that of (DiMarco. et al., 1993), which also uses the LOOM classifier, but with the complementary goal of handling fine-grained, stylistic variations. LOOM and other frame-based systems (e.g., KL-ONE and KRL) have also been used by a number of other researchers, including (Brachman and Schmolze, 1985), (MacGregor, 1991), (Woods and Brachman, 1978), among others. Alternative KR formalisms have been explored by a number of researchers including (Shapiro, 1993), (Iwanska, 1993), (Quantz and Schmitz, 1994), (Schubert, 1991), (Sowa, 1991). Most of these approaches have different objectives or address deeper conceptual issues. An example is the work of (Iwanska, 1993), which is concerned with notions such as logical inferences, entailment, negation, and quantification. The primary concern in Iwanska's work is the population of a knowledge base and the provision of a framework for truth maintenance and queries. While this work has certain elements in common with our approach (e.g., the representation of entailment, which is similar to our notion of classification), the framework is more applicable to a discourse analysis system than to the problem of lexical selection in a generation system.

The remainder of this section scopes out the space of possible lexical mismatches and describes our specific focus within this space.

8.2.1 Space of Lexical Mismatches

We have found that, although the problem of lexical mismatches has been characterized in a number of ways within the MT research literature, there does not exist an agreed-upon partitioning of the data in this *problem space.* Here we first take a brief look at a classic case of a lexical mismatch and a few ways of dividing up the problem space. We then present our own alternative approach to identifying mismatches.

Consider one classic mismatch case: Spanish has two words *pez* and *pescado* that correspond to the English word *fish. Pez* is used for a default generic, or unmarked, situation when it is assumed to include all fish,[29] and *pescado* is used in the specific, marked situation for a fish that has been caught and is no longer in its natural state.

The simplest **descriptive** characterization of this example is that it is a one-to-many mismatch from SL lexicon to TL lexicon. This frames the translation task as a problem of selecting one word out of a limited set of retrieved TL entries, a *listed lexical selection* problem. This classification of the example yields no useful generalizations however: different examples within the same class of "one-to-many" mismatches will require different MT solutions. Consider just one. The English word *know* translates into French *connaître* or *savoir*, a one-to-many mismatch that depends on the semantics of the verb's argument, not a limited selection at all. The one-to-many classification is too general, so that whatever MT solution is developed to handle the *fish/pez-pescado* case, it will have no bearing on this *know/connaître-savoir* case.

A **linguistic** characterization of this example is that it belongs to the narrower class of "unmarked-to-marked/unmarked" mapping mismatches, again from the SL to the TL lexicon. Note that operationally this presents a *lexical description* problem, identifying lexical entries as marked (leaving others unmarked) in order to address the listed lexical selection problem mentioned above. This classification of the example fails in another way, however; it is too narrow. The markedness information may be lexically ambiguous in one language but not in another. Consider *gato,* the Spanish word for both English words *cat* and *tom.* That is, *gato* is ambiguous with respect to markedness, being either unmarked for all cats or marked specifically for males cats.[30] To translate *gato* into English where two words are available, *cat* or *tom,* a MT system requires a *markedness preserving* mapping to ensure where possible that a marked word maps to a marked word and an unmarked word maps to an unmarked word.

Note that while this situation may appear to be the same as with translating *fish/pez-pescado*, it is not. With *fish,* there was no lexically stored ambiguity and so the mapping was "unmarked-to-marked/unmarked." In order to generate the marked case, a MT system must necessarily make use of knowledge available either from the rest of the sentence or from the larger context surrounding the sentence. This search will be driven by the TL lexicon in the generation phase. In the *gato* case, the markedness ambiguity is stored in the SL lexicon and so can be passed along during translation starting in the analysis phase. In short, even though partitioning the problem space using linguistic information gets us closer to understanding lexical mismatches, this approach combines the information that we need to distinguish between two distinct classes of MT problems.

---

[29] The marked/unmarked distinction is originally from phonology and later was extended into syntax, semantics and learnability theory (Bolinger, 1975).

[30] Markedness ambiguity is not language-specific. The English word *goose,* for example, is also ambiguous. It may be unmarked for sex, as in the sentence *That's a goose, not a chicken* or it may be marked as 'female', as in the sentence *That's a goose, not a gander.*

Given these examples, the challenge then is to determine which cases to group together into equivalence classes, so that a solution for one example in a partition or problem class will work on all the examples in that class. Our approach is **operational** and seeks to place the different mismatches where they are encountered in our MT system. We do not attempt here to cover the full space of lexical mismatches. Rather we have focused our research on individual partitions from those listed below. In our view of the space of mismatches we have restricted our attention to those that occur at the interface between the interlingua and the target language. We have left aside those that must be resolved by reference to broader contextual knowledge of transactions.[31] Here are a number of ways that the mapping from the interlingua to the target language may present options or problems for the MT system within a language:[32]

- synonymy
    - English *go by car, drive*
    - French *aller à pied, marcher* (go on foot, walk)
- gaps in lexicalization
    - no English word for *go by vehicle*
      but *bus, train, jet* (go by bus, train, jet)
    - no German word for *go*
      but *laufen, fahren* (go on foot, go by vehicle)
- core-overlap
    - English non-causative *fall* and causative *fell*
      but non-causative and causative *break*
    - German non-causative *fallen* and causative *fällen*
      but non-causative and causative *brechen*
- specialization/generalization
    - English *cook, bake, boil*
    - German *kochen, backen, sieden*
- core-overlap and
  specialization/generalization
    - English *move, take, steal*
    - French *bouger, prendre, voler*

---

[31] We will not discuss this second type of mismatch here. However we can give a short example that may give the reader a sense of what these mismatches entail. Consider the case of how the Japanese, French and Germans lexicalize the same transaction during a bus ride: *punch the ticket, validate the ticket,* and *invalidate the ticket* (Tsujii and Ananiadou, 1993). Each focuses on a different aspect of the overall transaction, one on the action itself (the punching), one on the state of the ticket during the ride (a valid ticket), and one on the state of the ticket after the ride (an invalid ticket). These are more properly labeled *transaction-focus mismatches.*

[32] Some of these examples are taken from related work by (DiMarco. et al., 1993). Their research addresses many of the same questions we are examining.

8.2.2 Our Focus

While all of the cases listed above have been relevant to our work building semantic ontologies, in the examples that follow we will focus on the lexicalization gaps in particular. We make the assumption that, for each SL word, there exists at least one TL word that is closest in meaning. From this it follows that, when an exact TL word match is missing (a gap), there are three possible relations between the closest TL word and the SL word: subsumes, subsumed-by, and overlapping.[33] These three lexical/ontological mismatches span the possible cases of meaning mismatch (Barnett et al., 1994). Here we will briefly give an example of each of these for translations from English into German.

In Figure 24 in the RHS column, we can see first that the non-causative English verb *bus* has no German lexical equivalent. However since German does have a slightly more general non-causative verb *fahren,* a selection algorithm can opt for a *subsumes* relation to resolve the mismatch and pick this TL word as the head for its translation of *bus.*[34]

A *subsumes* relation is not always available to resolve lexicalization gap mismatches. The non-causative English verb *move* is very general and does not exist as a lexical entry in German. At that level in the ontology, there is no more general non-causative concept to tap for the translation. In this case the selection algorithm must opt for a *subsumes-by* relation by picking a TL word below the SL word in the ontology. In translating verbs, this selection depends crucially on finding a TL verb whose constraints are met by both the SL verb and its arguments.

Finally an *overlap* relation occurs in translating the causative English verb *bus* where again we find no corresponding German lexical entry in the ontology. One option needed for translating this verb in a formal style of speech (such as in a legal document) involves decomposing its meaning into the comparable phrase *to cause to go by bus.* While the *cause* concept (for *veranlassen*) subsumes that of the causative *bus,* a reduction link relation (see lowest dashed arrow in Figure 24) — not a subsumption relation – is needed to capture *go by bus.*

These examples have been introduced here to clarify how lexicalization gaps fall into three lexical/ontological mismatch classes. We will come back to these examples in section 8.4 when we discuss the algorithm for traversing our semantic ontology in each of these cases.

8.3 Combining Aspects of Lexical Semantics and KB

Our research in developing a lexical semantics for MT is derived from the LCS formalism of (Jackendoff, 1983). His work however does not address the **computational** issues associated with representing or composing LCSs.[35] In particular, though Jackendoff writes that thematic relations (i.e. the roles in predicate-argument structure) depend crucially on an enriched ontology, he leaves open to interpretation (i) what that ontology or knowledge base ought to look like and (ii) what would constitute an adequate scheme for encoding the primitives of the LCS representation in the knowledge base. Our approach retains the benefits of the LCS formalism for lexical semantics while augmenting the representation with links into KB concepts. We explain these goals briefly below.

8.3.1 Benefits of Retaining the LCS formalism

Though several arguments have been made in favor of an LCS-based MT approach, we only cover two here; one benefit relates to IL MT systems in general, and the other bears specifically on capturing mismatches.

---

[33] The closest TL word is *over-general*, *over-specific*, or *overlapping* in meaning with respect to SL word's meaning.

[34] Suffice it to say here that the selected verb becomes the head of the TL phrase and the information dropped in the move up the ontological hierarchy becomes the restrictive modifier phrase to that head. This is the verb phrase analog to the approach taken by (Sondheimer et al., 1990) with noun phrases.

[35] He points this out explicitly in response to criticism from some in the computational linguistic community (Jackendoff, 1992).

One advantage to using an LCS-derived formalism for an interlingua is that its interface to the syntactic component of a MT system has been well developed. The LCS formalism provides a structured representation with predicate-argument forms and the potential for designating operator scoping relations. Because there is a "syntax" to these lexical-semantic representations, it is possible to provide a systematic mapping between the LCS representation and the corresponding syntactic structure. Furthermore, that mapping may be captured in a small set of linking rules, parameterized for cross-linguistic variation. (For an extensive description, see (Dorr, 1993b).)

The compositionality of this representation provides us with another benefit that is relevant to lexical selection research: the LCS formalism captures the phenomena of argument incorporation. Since languages differ with respect to what information they incorporate (Talmy, 1991), an IL formalism that retains incorporated information generation will provide a greater variety of lexicalization options at lexical selection time. In order to show how the LCS formalism can do this, we compare how three verbs that have different incorporation properties go through the IL composition and then the lexical selection phase during generation.

Informally incorporation refers to the semantics of one word, such as *lift,* hiding or incorporating the semantics of another word or phrase, such as *up.* Other words, such as *ascend* and *climb,* also contain within them the meaning *up.* This incorporation information is encoded in each verb's RLCS as the same substructure, the RLCS for the word *up.*

At analysis time the *up* substructures in *lift, ascend,* and *climb* are treated differently.[36] The *up* substructure inside of the *lift* RLCS is marked as optional and will absorb an identical structure during composition. When the RLCSs for *lift* and *up* are composed, the resulting composed form is identical to the RLCS for *lift* alone. By contrast, in the RLCS for *ascend,* the *up* substructure is unmarked and inaccessible during analysis.[37] Finally in the RLCS for *climb,* the *up* substructure is marked as a default option and it may be overwritten by non-identical structures during composition, as with *climb down.*

The markings on the RLCSs are removed after the analysis phase is completed, leaving a language-independent CLCS as the interlingual form in the MT system. The portion of the IL forms corresponding to (what was) *lift (up), ascend, climb (up)* all contain the same substructure *up.* In an MT system, the lexical generation step will decide whether or not to lexicalize the equivalent of a TL *up.*[38] The key benefit here is that the LCS formalism has preserved the substructure information, and so does not prematurely foreclose the decision process on lexicalization. The LCS formalism, by leaving the lexicalization decision open into the TL phase, allows for TL-specific pragmatic information to be used and for stylistic choices to be made in the final generation steps[39] — after the lexical options have been identified from the IL form via classification and the semantic ontology has been traversed.

8.3.2 Encoding the LCS formalism in the KB

We adopt the view that it is possible to consider the status of the LCS primitives from a KR point of view. We take the LCS primitives to be linguistic realizations of KR concepts, i.e., the LCS primitives are grounded in the KB. Part of the motivation for this combined framework is that it allows us to gain precision and accuracy with respect to the KB.

---

[36] If we were to classify these verbs in Figure 24, *lift* would be subsumed by [CAUSE(_ ,[GO-LOC(_ , _)])] and *ascend* and *climb* would be subsumed by [GO-LOC(_ , _)].

[37] We thereby avoid letting the RLCSs for *ascend* and *up* collapse into one RLCS for *ascend:* this lexical *up* does not have the spatial sense of the incorporated *up* substructures.

[38] If the target language were German, for either a *lift* or *lift up* SL input, there would be a *heben/anheben* choice (where the German prefix *an* corresponds loosely to the English *up* in this case). For either a *climb* or *climb up* SL input, there would be a *steigen/aufsteigen* choice (among several few others). And finally for an *ascend* SL input, there would be a *steigen/ersteigen* choice (among possibly a few others). These choices correspond to the synonymy mismatch in section 8.2.1.

[39] For example, the TL discourse may be informal and call for a *go up* in lieu of an *ascend.*

We achieve this LCS/KB melding by assuming (i) the base set of LCS primitives to be a base set of our LOOM-based *lcs-concepts* whose *lcs-roles* (if any) represent the positions in the LCS structures, and (ii) the non-primitive LCSs limited to those appearing as RLCSs in the MT lexicons to be another set of LOOM *lcs-concepts* defined recursively on the LOOM base set. In other words, for every LCS primitive and every non-primitive LCS found in the MT lexicons, there is an lcs-concept.

Recall that RLCSs are configurations of one or more of the LCS primitives composed into a single structure that meets the well-formedness conditions of the LCS "syntax." We may now restate the mapping from the last paragraph in terms of RLCSs: each primitive RLCS maps into one of the KB *base set* lcs-concepts and each non-primitive RLCS maps into one of the KB *derived set* lcs-concepts. Hence any relation that exists among substructures within an RLCS has a corresponding lcs-based relation among lcs-concepts in the KB. The full set of lcs-concepts and their lcs-based relations constitute the semantic ontology of the MT system encoded in the KB.

The next section provides further discussion on classifying RLCS-based lcs-concepts in the semantic ontology and on traversing the semantic ontology during translation runtime. The goal of the remainder of this section is to determine the mapping from relations among RLCSs[40] to relations among LOOM-based lcs-concepts.[41] In particular, we clarify why both subsumption and reduction links are needed in the KB to properly mirror the richness of relations among RLCSs.

Within a non-primitive RLCS, there are, as a result of the recursive definitions in the LCS syntax, two RLCS substructures that comprise C; one which we call the root RLCS (call it R) because it includes the root node, and the other which we call the non-root RLCS (call it N) that does not include the root.

When we map RLCSs into the KB we wish to preserve both the relation of C-to-R and the relation of C-to-N. A look at Figure 24 will help clarify that the relation of C-to-R is encoded in our semantic ontology as what we have been rather casually calling a *subsumption link.* For example, the RLCS for *move* in the left-hand column stands in a C-to-R relation with the RLCS for *cause.* We capture this in the KB by saying that *move* is subsumed by *cause.* The relation of C-to-N is encoded as a *reduction link.* For example, the RLCS for the causative *move* stands in a C-to-N relation with the RLCS for the non-causative *move;* the non-causative *move* is reached via a reduction link from the causative *move.* As discussed in the next section, the reduction links serve to partition the lcs-concept being translated into subcomponents where each has its meaning preserved.

8.4 LOOM Implementation

There are two stages to our solution to the lexical mismatch problem. The first involves the classification of words according to their conceptual description prior to translation runtime. The second involves the selection of words during translation runtime based on the conceptual representation that comprises the IL representation. Each of these will be described, in turn.

8.4.1 Before Lexical Match Time

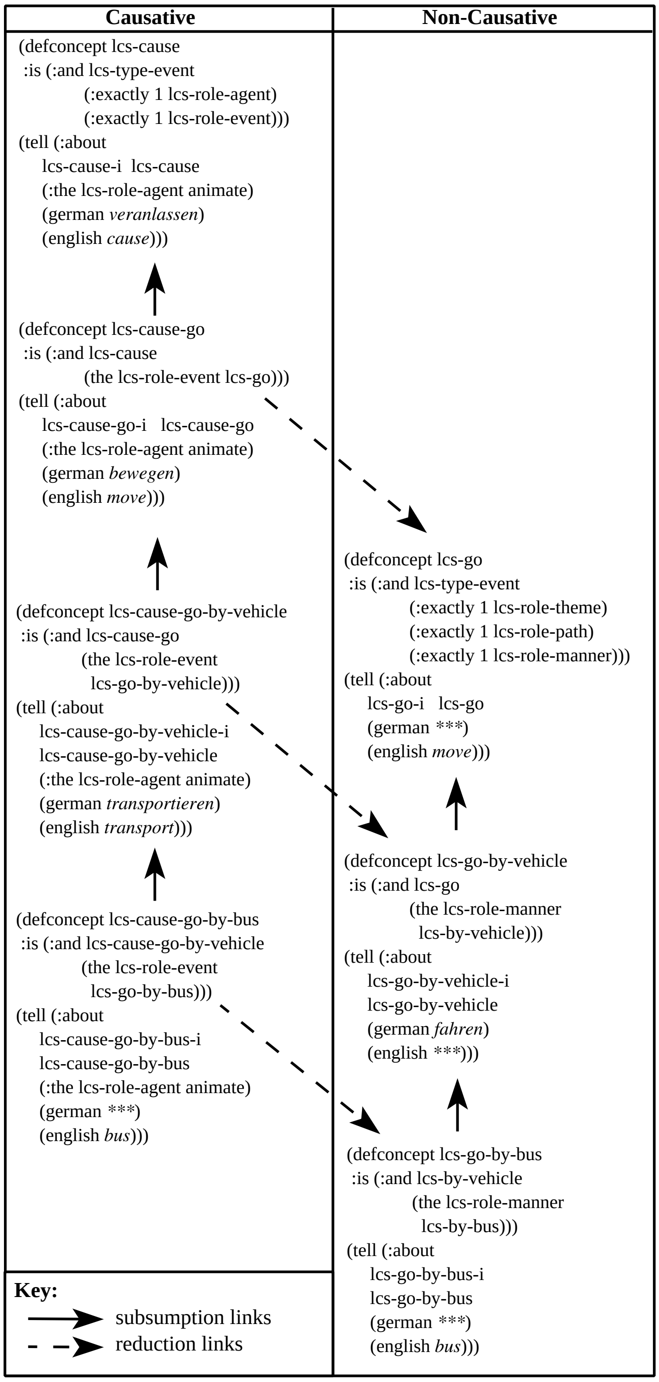
Our approach requires the semantic lexicographer to create conceptual descriptions (RLCS's) embodying the meaning of lexical entries.[42] As discussed earlier, these representations are stored in their own language-specific lexicons. When the RLCS definitions are loaded, LOOM automatically

---

[40] Here we use RLCS to refer to the Jackendoffian encoding of lexical entries as seen in simplified form in Figure 24.

[41] Here we use lcs-concepts to refer to the LOOM-based encoding of RLCSs.

[42] We currently have a database containing 250 RLCS templates for English typed in by hand on the basis of work by (Levin, 1993). We expect to automatically acquire 3200 English verbs using these templates as mentioned above in section 7. Once this acquisition process is complete, we are planning to scale up the current experiment so that we can classify this larger set in terms of the LOOM representation.



```
(defconcept lcs-cause
 :is (:and lcs-type-event
           (:exactly 1 lcs-role-agent)
           (:exactly 1 lcs-role-event)))
(tell (:about
    lcs-cause-i  lcs-cause
    (:the lcs-role-agent animate)
    (german veranlassen)
    (english cause)))
```

```
(defconcept lcs-cause-go
 :is (:and lcs-cause
           (the lcs-role-event lcs-go)))
(tell (:about
    lcs-cause-go-i   lcs-cause-go
    (:the lcs-role-agent animate)
    (german bewegen)
    (english move)))
```

```
(defconcept lcs-cause-go-by-vehicle
 :is (:and lcs-cause-go
           (the lcs-role-event
             lcs-go-by-vehicle)))
(tell (:about
    lcs-cause-go-by-vehicle-i
    lcs-cause-go-by-vehicle
    (:the lcs-role-agent animate)
    (german transportieren)
    (english transport)))
```

```
(defconcept lcs-cause-go-by-bus
 :is (:and lcs-cause-go-by-vehicle
           (the lcs-role-event
             lcs-go-by-bus)))
(tell (:about
    lcs-cause-go-by-bus-i
    lcs-cause-go-by-bus
    (:the lcs-role-agent animate)
    (german ***)
    (english bus)))
```

```
(defconcept lcs-go
 :is (:and lcs-type-event
           (:exactly 1 lcs-role-theme)
           (:exactly 1 lcs-role-path)
           (:exactly 1 lcs-role-manner)))
(tell (:about
    lcs-go-i   lcs-go
    (german ***)
    (english move)))
```

```
(defconcept lcs-go-by-vehicle
 :is (:and lcs-go
           (the lcs-role-manner
             lcs-by-vehicle)))
(tell (:about
    lcs-go-by-vehicle-i
    lcs-go-by-vehicle
    (german fahren)
    (english ***)))
```

```
(defconcept lcs-go-by-bus
 :is (:and lcs-by-vehicle
           (the lcs-role-manner
             lcs-by-bus)))
(tell (:about
    lcs-go-by-bus-i
    lcs-go-by-bus
    (german ***)
    (english bus)))
```

Figure 25: LOOM Version of RLCS Classification

infers the ontological relationships between their concepts, thus creating a hierarchy[43] into which each language's lexical entries point. Figure 25 illustrates the LOOM specification of the ontology presented earlier in Figure 24.

Note that lcs-roles in Figure 25 correspond to thematic positions (underscores) in Figure 24. Concepts and their language-specific instances are prefixed by "lcs-"; the instances are suffixed by "-i". The RLCSs of the language-specific lexicons in Figure 24 and their LOOM-encoded versions of the semantic ontology in Figure 25 have been simplified here for ease of presentation.

8.4.2 Lexical Selection and Mismatch Handling

This section contains a functional description of the lexical selection process and some associated mismatch handling that the TL classification hierarchy buys us. For the inner workings of the algorithm, see section 8.4.3 below. We discuss the three mismatch cases mentioned earlier in section 8.2.2. (In this context "traversing" will refer to the search procedure at translation runtime that is invoked following the initial mismatch diagnosis returned by the term classifier.)

For the purpose of this discussion, it will suffice to focus on the translation of single words rather than an entire sentence. This process is intended to operate recursively, i.e., once the TL verb is selected the same lexical selection procedure is applied to the arguments of that word.

Resolution of Subsumption Cases:

From a high level view, the treatment of both the *subsumes* and the *subsumed-by* cases are similar. Both cases produce lists of word concepts: the first list simply contains the ontological parents of the CLCS we look up in the TL hierarchy, while the second list contains the ontological children. The subsumed-by case corresponds to a situation where the closest matching TL word is subsumed by the SL word concept. The subsumes case corresponds to a situation where the closest matching TL word subsumes the SL word concept.[44]

Consider first the non-causative English verb *bus* in figure 24, as used in the sentence *they bussed into the city.* As mentioned earlier, there does not exist an equivalent lexical entry among the verbs in German (which we have annotated in Figure 25 with a '***'). The best translation, *Sie fuhren mit dem Bus in die Stadt,* uses the combination of the verb *fahren* (go by vehicle) and the prepositional phrase *mit dem Bus* (by bus) to convey a comparable meaning. In our interlingual representations, this mapping to the German verb *fahren* from the English verb *bus* corresponds to the subsumes case: the TL meaning [GO-LOC(_ , _), (BY VEHICLE)] subsumes the SL meaning [GO-LOC(_ , _), (BY BUS)]. In the semantic ontology, this corresponds to a straightforward traversal step upward along a subsumption link. Note that although this link is within the ontology constructed with LCS concepts), it is grounded in the general, i.e. non-LCS ontology, part of the KB where the concept vehicle subsumes the concept bus.

Next consider the non-causative English verb *move,* as used in the sentences (i) *the tree moved* and (ii) *the car moved through the tunnel.*[45] This very general sense of *move,* which we encode as GO-LOC in the RLCS to indicate a change of spatial location, does not translate directly into German. At this level in the ontology, there is no more general non-causative concept to tap for the translation. Consequently the best translations for (i) and (ii) respectively, *der Baum bewegte sich* and *das Auto fuhr durch den Tunnel,* make use of less general verbs that, by definition, capture a narrower meaning. With *sich bewegen* the nature of the movement is constrained, and with *fahren* the logical subject of the movement is constrained.[46] These mappings to *sich bewegen/fahren* from

---

[43] We use the term *hierarchy*, in a loose sense which does not exclude multiple inheritance.

[44] In LOOM, we use the commands direct-superconcepts and direct-subconcepts for reaching immediate ancestors and descendants, respectively, in combination with the command superconcepts to test for other adequate TL word concepts.

[45] We identify these usages of the word *move* as non-causative in the broadest sense that the cause or causer of the movement is unknown.

[46] In terms of figure 25, *sich bewegen,* were it included, would be a child of the lcs-go concept.

move correspond to a traversal step down a subsumption link from the SL word concept in the ontology.[47]

Resolution of Overlapping Cases:

By far, the most interesting case is that of the *overlapping* case. Whereas the subsumes and subsumed-by cases are covered by the Directed Subsumption Links, the overlapping cases require the use of Crossover Reduction Links, which constitute the main contribution of our representation scheme.

This case is more complex that the others in that the TL and SL meanings may overlap in several ways, some of which cannot be classified strictly in terms of subsumption. Several translations are possible, reflecting these many overlapping combinations. As mentioned earlier, the English causative verb *bus,* as used in the sentence *the parents bussed the children to school,* has no German lexical equivalent. One German translation of this sentence that conveys an official or formal style is *die Eltern veranlassten die Kinder, mit dem Bus zur Schule zu fahren* (the parents caused the children to go to school by bus). In our interlingual representations, the meanings for the German words *veranlassen* [CAUSE(_ , _)] and *fahren* [GO(_ , _), (BY VEHICLE)] each overlap the English meaning for *bus* [CAUSE(_ ,[GO-LOC(_ , _),(BY BUS)])]. (See lower left corner of figure 24.) While the causative *bus* is subsumed by *veranlassen,* there is a reduction link relation in the traversal from *bus* to *fahren.*

Note that there often arise several possible translations in the overlapping cases precisely because there are several ways to lexicalize this overlapping concepts. The search procedure that traverses the ontology explores several paths trying to find the set of matches from the German lexicon that covers the IL form. In general, reduction link traversals are preferred over subsumption link traversals since reduction links preserve exact substructures whereas subsumption links require inferencing.

8.4.3 Inner Workings of the Lexical Selection Algorithm

A key point about the algorithm is that the classifier, which is used prior to processing time, is also a main driving component of the lexical selection scheme.

The details are given here:

- **CLCS Lookup:** Search the ontology on the basis of the IL representation (i.e., the CLCS of (Dorr, 1993b)), in order to determine its hierarchical position[48]
- **RLCS Extraction:** Examine instances of TL words that occur at the current hierarchical position.
- **Merge Detection:** In the cases where there exists an exact match between a TL RLCS and the IL structure, the associated lexical item is returned.[49]
- **Traversal of Semantic Ontology:** In the cases where there is no exact match between a TL RLCS and the IL structure, a search procedure is invoked to traverse the LCS concept ontology, following subsumption and reduction links. Along each search path, after traversing each link, evaluate returned TL RLCSs for coverage of IL form.

---

[47] At this stage in generation, we are concerned with extracting lexicalization options, building a set of one or more RLCSs in the TL. At the subsequent step of TL RLCS composition, the constraints defined within each LOOM-version of the TL RLCS will be checked against each other.

[48] When the CLCS corresponds to a SL word's RLCS, the search will find a match in the ontology corresponding to the concept where the SL was classified before translation runtime. What is at issue in the next step is whether, for this SL concept just looked up, the TL will have a word defined as an instance off of that same concept.

[49] This was the case for the example given in section 8.1 translating the German *bewegen* into English *move.*

- **Overlapping Case:** The reduction links are followed in order to find a set of possible substructures that provide full coverage for the concept. Return a list of associated reduction RLCSs.
- **Familial Information Case:** Subsumption links are followed, upward by default, downward if this is not possible because traversal is at top of LCS subsumption hierarchy. Return the list of ontological parents (if upward traversal) or list of ontological children (if downward traversal).

Note that overlap reduction has a higher standing than subsumption. This is because our goal is to get the fullest possible coverage of the IL concept. With reduction, we only partition the IL form in terms of its coverage by a TL RLCS. With subsumption, we may lose information in the inferencing that would have led to a more accurate lexicalization.

8.5 Areas for Future Investigation: KB/IL

Future work will include the detection of non-subsumption/overlap cases, i.e., where differences are not resolved solely by means of appropriate word selection but rather by means of constraints on relations defined by the roles encoded in word definitions. For example, one could envision a language where the concept *steal* could only be realized as some combination of the concepts *take* and *unauthorizedly*. Suppose these were classified in the ontology, but that there were no word for the concept *unauthorizedly*. Constraints on the relations between roles would allow us to translate *unauthorizedly* as something akin to *without permission*, perhaps as an instantiation of the concept *movement of an item by a person that is not a member of the item's list of authorized custodians*.

An additional area for future investigation is the use of knowledge within a KB to assist in filtering. For example, if we were translating the sentence *Jane bought a fish from the pet shop* into Spanish, we would need to have knowledge about pet shops, i.e., that a pet shop sells pets and supplies. With deeper knowledge, the word *pescado* would be filtered out because the associated concept (a fish that has been caught to be eaten) would not unify with purchase activities entailed by a pet shop.

9 Relation of LCS to TAGs

This section describes current collaboration with Dr. Martha Palmer at the University of Pennsylvania in which we have examined the construction of a lexicon for interlingual machine translation and also the analysis of formal properties of an interlingua as a language in its own right. As such it should be possible to define a lexicalized grammar for the representation of lexical entries and a set of operations over that grammar that can be used to both analyze and generate interlingua representations. The interlingua we discuss in this paper is Lexical Conceptual Structure (LCS) as formulated by (Dorr, 1993b). The next section presents a grammar for LCS as a representation language. The grammar formalism whose operations we examine with respect to their ability to compose LCS representations is Feature-Based Lexicalized Adjoining Grammar, (FB-LTAG), a version of Tree Adjoining Grammar (TAG) (Joshi et al., 1975), (Schabes, 1990), (Vijay-Shanker, 1987) and its description, along with example TAG structures, forms our final section. What we find is that the implementation of LCS as a TAG, although not completely straightforward, can be done, providing the full power of the well-defined mathematical properties of TAGs as a basis for describing the formal properties of LCS.

9.1 Grammar for LCS

The current task is to explore the LCS representation in the context of an FB-LTAG model in order to test hypotheses about the interlingual representation for machine translation. Our goal

```
Situation →
  LET {Thing, Event, State} {Event, State} |
  CAUSE {Thing, Event, State} {Event, State} |
  GO {Thing, Event, State} Path |
  GO-EXT {Thing, Event, State} Path |
  ORIENT Thing Path |
  STAY {Thing, Event, State} Position |
  BE {Thing, Event, State} Position
Path →
   {TO, TOWARD, FROM, AWAY-FROM, VIA}
   {Thing, Event, State}
   Position
Position →
   {AT, IN, ON, ...}
   {Thing, Event, State}
   {Thing, Event, State, Property, Time}
Thing →
   {BOOK, PERSON, ROOM, ...}
Time →
   {TODAY, SATURDAY, 2:00, 4:00, ...}
Property →
   {TIRED, HUNGRY, RED, ...}
```

Figure 26: LCS Wellformedness Conditions Expressed as a Context-Free Grammar

is to develop a framework within which we can evaluate, formally, the expressive power of the representation language used in the lexicon, and also to determine systematically the depth of coverage with respect to different cross-linguistic phenomena.

In order to employ the TAG formalism, we must first associate a "syntax" with our "semantics." That is, we must express the wellformedness conditions on the LCS representation in terms of a "grammar," analogous to a context-free description at the level of syntactic structure. (See Figure 26.) In this grammar, curly brackets {} correspond to a choice of one, and only one, item. An example of a Path LCS would be the primitive TO with an Event and a Position as its two "arguments." Primitives correspond to the terminal nodes of the grammar (and are written in all capital letters); Types correspond to the nonterminal nodes of the grammar (and are written in lower case with an initial capital letter). Note that there are closed-class primitives (e.g., Situations and Paths) and open-class primitives (e.g., Things and Properties). There are also primitives which represent a large, but finite set (e.g., Positions).

Superimposed on this grammar is a set of wellformedness conditions corresponding to the "Field" mentioned in the previous section. In the FB-LTAG framework, the Field is not specified in terms of grammar rules, but is available by means of a feature specification. The feature ensures that Locational GO primitives only take Locational Paths, for instance. The full set of wellformedness conditions is as shown in Figure 27.

9.2 Tree Adjoining Grammar (TAG)

FB-LTAG is a version of Tree Adjoining Grammar (TAG) (Joshi et al., 1975), (Schabes, 1990), (Vijay-Shanker, 1987), that has been extended to include lexicalization and unification-based feature structures. In a TAG there are two types of elementary trees: initial trees and auxiliary trees.

Field (Feature)	**Arguments**
Locational	Arg 1: {Thing, Event, State}
	Arg 2: {Thing, Event}
Temporal	Arg 1: {Event, State}
	Arg 2: {Event, State, Time}
Identificational	Arg 1: Thing
	Arg 2: {Thing, Event, Property}
Possessional	Arg 1: {Thing, Event, State}
	Arg 2: Thing
Instrumental	Arg 1: {Event, State}
	Arg 2: Thing
Perceptual	Arg 1: Thing
	Arg 2: {Thing, Event, State}
Circumstantial	Arg 1: Thing
	Arg 2: {Event, State}
Intentional	Arg 1: {Event, State}
	Arg 2: {Thing, Event, State}
Existential	Arg 1: {Thing, State}
	Arg 2: EXIST

Figure 27: Wellformedness Conditions on LCS Fields

The frontier of an **initial tree** has as its anchor a terminal; the rest of the nodes on the frontier are non-terminals marked as substitution nodes. In addition to an anchor, and possible non-terminal substitution nodes, an **auxiliary tree** is required to have one node on the frontier marked as the foot node. The foot node must have the same category label as the tree's root node.

From a linguistic perspective, the set of elementary trees anchored by a lexical item represent the item's possible subcategorization frames. In an FB-LTAG, each lexical item is associated with a set of elementary trees, for which it is the lexical **anchor**. Each node in the tree has two sets of feature structures, the TOP and the BOTTOM. The BOTTOM feature structure contains information relating to the subtree rooted at the node, and the TOP feature structure contains information relating to the supertree at that node. Substitution nodes have only a TOP feature structure, while all other nodes have both a TOP and BOTTOM feature structure. Trees can be composed by applying two operations, substitution and adjunction, as shown in Figure 28.[50]

For **substitution** to occur, there must be a non-terminal frontier node marked for substitution in an elementary tree and a corresponding elementary tree whose root has the same label as that node (Figure 28(a)). Then the substitution node is replaced by the corresponding elementary tree. Substitution only operates on the frontier of a tree. **Adjunction**, on the other hand, can operate on an internal node, actually inserting an auxiliary tree at that point (Figure 28(b)). For this to occur, the internal node in the first tree must have the same label as both the root node and the foot node of the tree being adjoined onto it. The TOP feature structure of the internal node unifies with the TOP feature structure of the root node, and the BOTTOM feature structure unifies with the BOTTOM feature structure of the foot node. For linguistic reasons, initial trees are non-recursive tree structures, whereas auxiliary trees used for adjunction are required to be recursive. For the final tree to be valid all substitution nodes must be filled, and the TOP and BOTTOM feature

---

[50] Technically, substitution is a specialized version of adjunction, but it is useful to make a distinction between the two. These figures are used by permission from (XTAG, 1995). Abbreviations in the tree figure: t=top feature structure, b=bottom feature structure, tr=top feature structure of the root, br=bottom feature structure of the root, tf=top feature structure of the foot, bf=bottom feature structure of the foot, $\cup$ =unification.

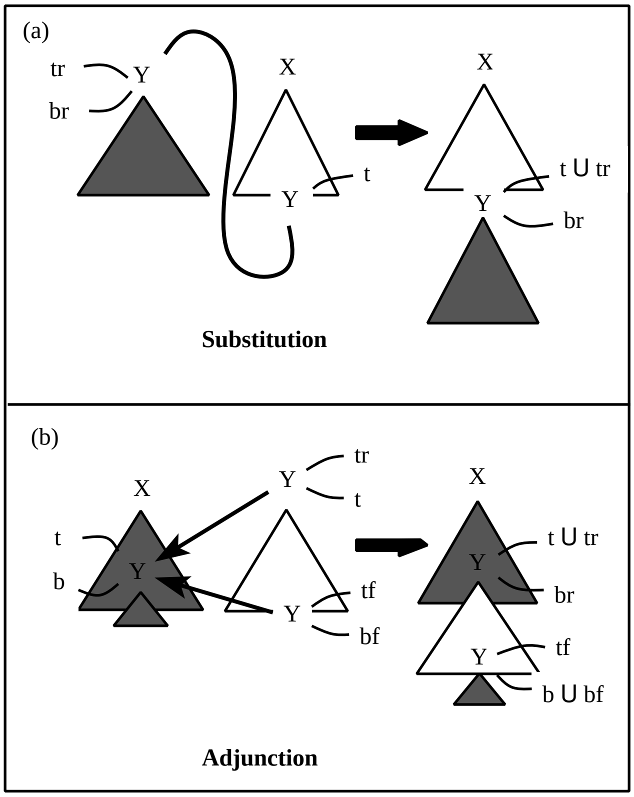


Figure 28: Substitution and Adjunction in FB-LTAG

structures on each node must unify with each other.

9.3 LCS Operation Requirements

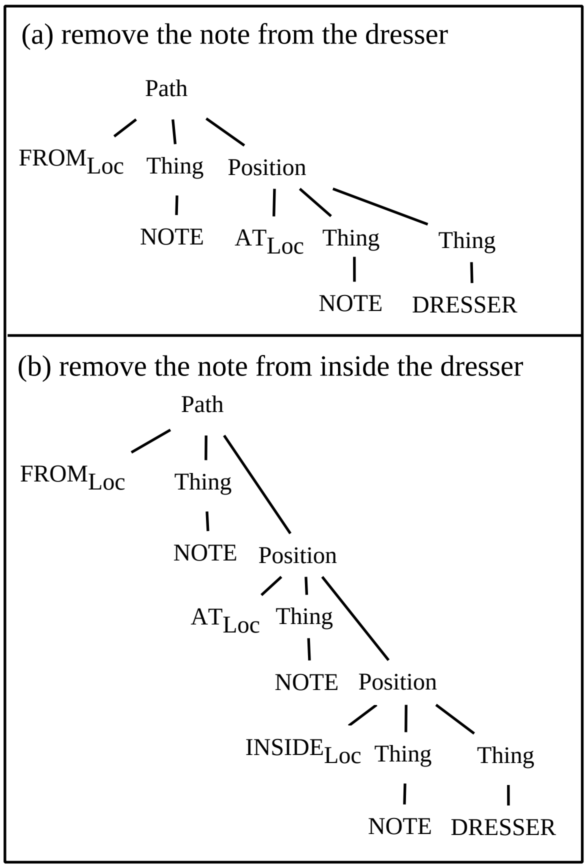
LCS structures that correspond to different syntactic elements of a sentence are composed to form the complete sentence representation. For instance, a string of prepositional phrases such as *over the hill, behind the stream, next to the woods* results in a recursive embedding of several path and position predicates. There is a clearly defined relationship between the representation of the phrase, *from the inside of the dresser* in *remove the note from the inside of the dresser*, whose LCS is given in Figure 29(a), and the simplified version, *from the dresser* given in Figure 29(b).

At first glance, that relationship would appear to be the TAG adjunction operation. However, the necessary conditions for adjunction are not met because there is not a recursive grammar rule for Position that allows a new Position node to be adjoined underneath the AT. We add a new grammar rule that provides this recursive definition:

Position →
{AT, IN, ON, ...}
{Thing, Event, State}
Position

Then, given the trees in Figure 30(a) and (b), adjunction can be applied to produce the tree in Figure 30(c).

The LCS *overlap* operation is more problematic. In the LCS representation of the sentence, *John lifted Mary up to the table,* there is a duplication between the UP component of LIFT, and the LCS representation of the UP TO prepositional phrase. The overlap for this example or other similar examples does not conform to adjunction as described here, since more than a single node is duplicated in both trees. It is necessary either for these duplicated nodes to be effectively merged or



(a) remove the note from the dresser

(b) remove the note from inside the dresser

Figure 29: LCS Representations for *from* vs. *from inside*

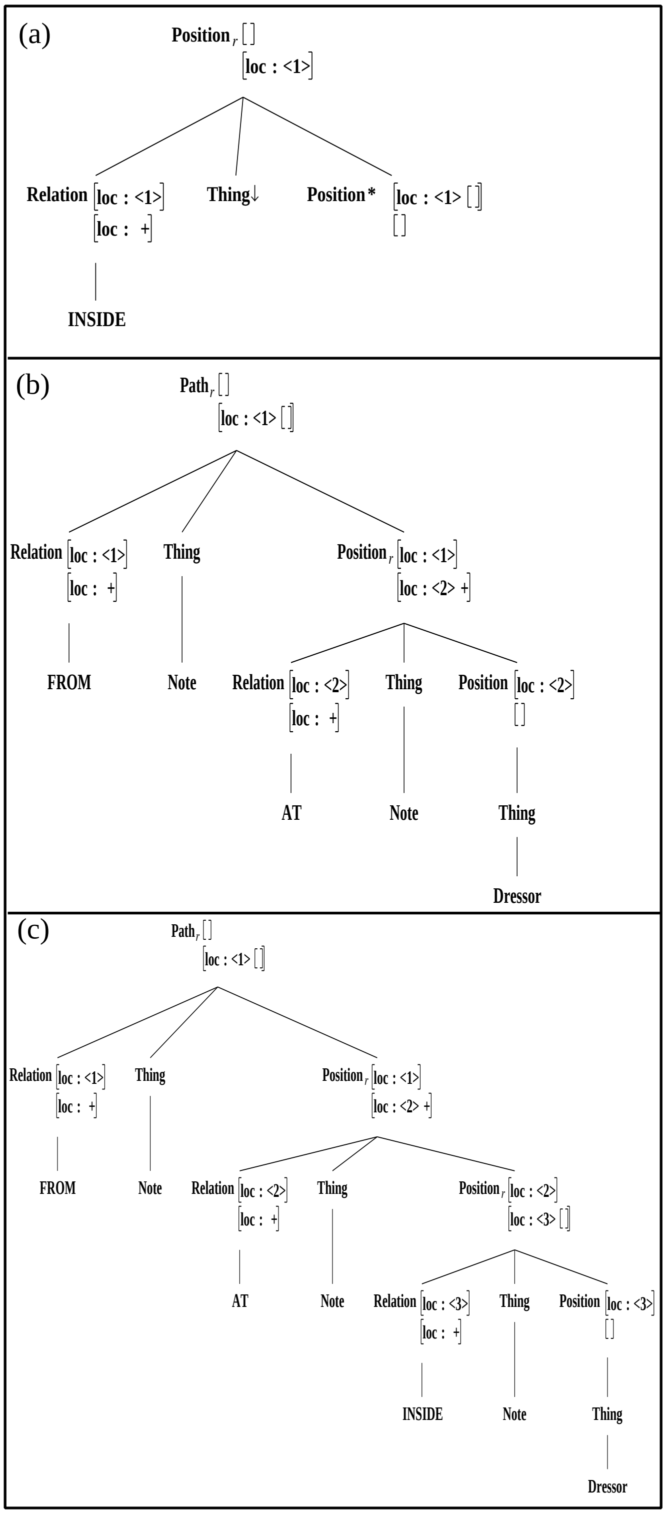


Figure 30: TAG Trees for *inside*, *from*, and *from inside*

to find another way of representing the information. The cleanest alternative from the perspective of the TAG formalism is to represent the UP component of LIFT in the feature structure as an overridable default. In this case, when the LCS structure corresponding to an UP TO prepositional phrase is being adjoined, also with an UP in the feature structure, the two UPs will unify. If a different type of LCS structure needs to be adjoined, such as the structure corresponding to the DOWN prepositional phrase in *The mother lifted the child down from the carousel horse,* then the UP feature can be overridden. If there is no prepositional phrase specified, then the feature still contains the information that the direction is inherently in an UP direction. There may be particular examples where incorporating the required default information as a feature is counter-intuitive. In that case, another possibility which does not require any altering of the LCS structure would be to use partial descriptions of trees for the prepositional phrases, with multi-component adjunction so that they can be adjoined onto the initial tree as described in (Vijay-Shanker, 1992).

9.4 Implications and Future Directions

We have described certain aspects of using the TAG formalism for the implementation of LCS as an interlingua. The standard operations of substitution and adjunction apply, and they can be extended to handle the overlap LCS operation. This gives us a formal structure with well-defined operations that imposes constraints on the composition of LCS, aiding in the regularization of LCS procedures. More importantly, it opens up the possibility of using the well-known mathematical properties of the TAG formalism to prove properties about LCS as an interlingua.

As discussed by (Dorr and Voss, 1993), machine translation theory has not yet addressed the issues surrounding how the interlingua of a MT system should be defined or evaluated. We believe that the investigation described above is the first step toward providing a framework in which MT developers can define and evaluate different lexical representations with respect to coverage and efficiency.

References

Abney, S. (1989). A computational model of human parsing. *Journal of Psycholinguistic Research*, 18(1):129–144.

Abney, S. and Cole, J. (1985). A government-binding parser. In *Proceedings of NELS, No. 16*, pages 1–17.

Alshawi, H. (1989). Analysing the dictionary definitions. In Boguraev, B. and Briscoe, T., editors, *Computational Lexicography for Natural Language Processing*, pages 153–169. Longman, London.

Ballard, B. and Stumberger, D. (1986). Semantic acquisition in teli. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics.*

Barnett, J., Knight, K., Mani, I., and Rich, E. (1990). Knowledge and natural language processing. *Communications of the ACM.*

Barnett, J., Mani, I., and Rich, E. (1994). Reversible machine translation: What to do when the languages don't match up. In Strzalkowski, T., editor, *Reversible Grammar in Natural Language Processing.* Kluwer Academic Publishers.

Barton, G., Berwick, R., and Ristad, E. (1987). *Computational Complexity and Natural Language.* MIT Press, Cambridge, MA.

Bates, M. and Bobrow, R. (1983). A transportable natural language interface. In *Proceedings of the Sixth Annual ACM SIG Conference on Research and Development in Information Retrieval.*

Beaven, J. (1992). Shake and bake machine translation. In *Proceedings of Fourteenth International Conference on Computational Linguistics*, pages 603–609, Nantes, France.

Billot, S. and Lang, B. (1989). The structure of shared forests in ambiguous parsing. In *Proceedings of ACL–89*, pages 143–151.

Boguraev, B. and Briscoe, T. (1989). Utilising the ldoce grammar codes. In Boguraev, B. and Briscoe, T., editors, *Computational Lexicography for Natural Language Processing*, pages 85–116. Longman, London.

Bolinger, D. (1975). *Aspects of Language.* Harcourt Brace Jovanovich, New York, NY.

Brachman, R. and Schmolze, J. (1985). An overview of the kl-one knowledge representation system. *Cognitive Science*, 9(2):171–216.

Choi, S. and Bowerman, M. (1991). Learning to express motion events in english and korean: The influence of language-specific lexicalization patterns. In Levin, B. and Pinker, S., editors, *Lexical and Conceptual Semantics*, pages 193–228. Elsevier, Amsterdam.

Chomsky, N. (1981). *Lectures on Government and Binding.* Foris Publications, Cinnaminson, USA.

Chomsky, N. (1986a). *Barriers.* MIT Press, Cambridge, MA.

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

Corbin, W., Copeland, D., and Buck, B. (1994). Determining verb usage from parsed corpora: Matrix of levin's syntactic/semantic classes. Technical Report Project Report for NLP Course (CMSC 723), University of Maryland, College Park, MD.

Correa, N. (1991). Empty categories, chains, and parsing. In Berwick, R., Abney, S., and Tenny, C., editors, *Principle-Based Parsing: Computation and Psycholinguistics*, pages 83–121. Kluwer Academic Publishers.

Cottrell, G. (1989). *A Connectionist Approach to Word Sense Disambiguation.* Morgan Kaufmann, Los Altos, CA.

DiMarco., C., Hirst, G., and Stede, M. (1993). The semantic and stylistic differentiation of synonyms and near-synonyms. Technical Report AAAI-93 Spring Symposium on Building Lexicons for Machine Translation, Stanford University, Stanford, CA.

Dorr, B. (1990). Solving thematic divergences in machine translation. In *Proceedings of ACL–90*, pages 127–134, University of Pittsburgh, Pittsburgh, PA.

Dorr, B. (1991). Principle-based parsing for machine translation. In Berwick, R., Abney, S., and Tenny, C., editors, *Principle-Based Parsing: Computation and Psycholinguistics*, pages 153–184. Kluwer Academic Publishers.

Dorr, B. (1992). The use of lexical semantics in interlingual machine translation. *Machine Translation*, 7(3):135–193.

Dorr, B. (1993a). Interlingual machine translation: a parameterized approach. *Artificial Intelligence*, 63(1&2):429–492.

Dorr, B. (1993b). *Machine Translation: A View from the Lexicon.* MIT Press, Cambridge, MA.

Dorr, B. (1994). Machine translation divergences: A formal description and proposed solution. *Computational Linguistics*, 20(4):597–633.

Dorr, B. and Voss, C. (1993). Machine translation of spatial expressions: Defining the relation between an interlingua and a knowledge representation system. In *Proceedings of AAAI–93.*

Egedi, D., Palmer, M., Park, H., and Joshi, A. (1994). Korean to english translation using synchronous tags. In *Proceedings of the First Conference of the Association for MT in the Americas*, Columbia, MD.

Farwell, D., Guthrie, L., and Wilks, Y. (1975). Automatically creating lexical entries for ultra, a multilingual mt system. *Machine Translation*, 8(3).

Fillmore, C. (1968). The Case for Case. In Bach, E. and Harms, R., editors, *Universals in Linguistic Theory*, pages 1–88. Holt, Rinehart, and Winston.

Fong, S. (1991). The computational implementation of principle-based parsers. In Berwick, B., Abney, S., and Tenny, C., editors, *Principle-Based Parsing: Computation and Psycholinguistics*, pages 65–82. Kluwer Academic Publishers.

Frank, R. (1990). Licensing and tree adjoining grammar in gb parsing. In *Proceedings of ACL–90*, pages 111–118, University of Pittsburgh, Pittsburgh, PA.

Ginsparg, J. (1983). A robust portable natural language data base interface. In *Proceedings of the Conference on Applied Natural Language Processing.*

Grimshaw, J. (1990). *Argument Structure.* MIT Press, Cambridge, MA.

Grishman, R. (1986). *Computational Linguistics.* New York, Cambridge University Press.

Grosz, B., Appelt, D., Martin, P., and Pereira, F. (1987). Team: An experiment in the design of transportable natural-language interfaces. *Artificial Intelligence*, 32(2):173–243.

Gruber, J. (1965). *Studies in Lexical Relations.* PhD thesis, MIT, Cambridge, MA.

Guida, G. and Tasso, C. (1983). Ir-nl1: An expert natural language interface to online data bases. In *Proceedings of the Conference on Applied Natural Language Processing.*

Haegeman, L. (1991). *Introduction to Government and Binding Theory.* Basil Blackwell Ltd.

Hogan, C. and Levin, L. (1994). Data sparseness in the acquisition of syntax-semantics mappings. In *Proceedings of the Post-COLING94 International Workshop on Directions of Lexical Research*, pages 153–159, Nicoletta Calzolari and Chengming Guo (co-chairs), Tshinghua University, Beijing.

Hornstein, N. (1990). *As Time Goes By.* MIT Press, Cambridge, MA.

Ihm, Hong, and Chang (1988). *Korean Grammar.*

Iwanska, L. (1993). Logical reasoning in natural language: It is all about knowledge. *International Journal of Minds and Machines, Special Issue on Knowledge Representation for Natural Language*, 3:475–510.

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

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

Jackendoff, R. (1992). What is semantic structures about. *Computational Linguistics*, 18(2):240–242.

Jones, M. (1987). Feedback as a coindexing mechanism in connectionist architectures. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 602–610.

Joshi, A., Levy, L., and Takahashi, M. (1975). Tree adjunct grammars. *Journal of Computer and System Sciences*.

Kameyama, M., Ochitani, R., Peters, S., and Sirai, H. (1991). Resolving translation mismatches with information flow. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 193–200, University of California, Berkeley, CA.

Kempen, G. and Vosse, T. (1989). Incremental syntactic tree formation in human sentence processing: A cognitive architecture based on activation decay and simulated annealing. *Connection Science*, 1(3):273–290.

Kinoshita, S., Phillips, J., and Tsujii, J. (1992). Interaction between structural changes in machine translation. In *Proceedings of Fourteenth International Conference on Computational Linguistics*, pages 679–685, Nantes, France.

Knight, K. (1991). *Integrating Knowledge Acquisition and Language Acquisition.* PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Lee, Y.-S. (1993). *Scrambling as Case-Driven Obligatory Movement.* PhD thesis, University of Pennsylvania, Philadelphia, PA.

Lenat, D. and Guha, R. (1990). *Building Large Knowledge-Based Systems.* Reading, MA, Addison-Wesley.

Levin, B. (1993). *English Verb Classes and Alternations: A Preliminary Investigation.* Chicago, IL.

Levin, L. and Nirenburg, S. (1993). Principles and idiosyncracies in mt lexicons. In *Working Notes for the AAAI Spring Symposium on Building Lexicons for Machine Translation*, pages 122–131, Stanford University, CA.

Lin, D. (1993). Principle-based parsing without overgeneration. In *Proceedings of ACL–93*, pages 112–120, Columbus, Ohio.

Lin, D. and Goebel, R. (1993). Context-free grammar parsing by message passing. In *Proceedings of PACLING–93*, Vancouver, BC.

Lindop, J. and Tsujii, J. (1991). Complex transfer in mt: A survey of examples. Technical Report CCL/UMIST Report 91/5, Center for Computational Linguistics, UMIST, Manchester, UK.

MacGregor, R. (1991). The evolving technology of classification-based knowledge representation systems. In Sowa, J., editor, *Principles of Semantic Networks*, pages 385–400. Morgan Kaufmann, San Mateo, CA.

Melby, A. (1986). Lexical transfer: Missing element in linguistic theories. In *Proceedings of Eleventh International Conference on Computational Linguistics*, Bonn, Germany.

Mitamura, T. (1990). *The Hierarchical Organization of Predicate Frames for Interpretive Mapping in Natural Language Processing.* PhD thesis, Department of Linguistics, University of Pittsburgh, Pittsburgh, PA.

Montemagni, S. and Vanderwende, L. (1992). Structural Patterns vs. String Patterns for Extracting Semantic Information from Dictionaries. In *Proceedings of the Fourteenth International Conference on Computational Linguistics*, pages 546–552, Nantes, France.

Nirenburg, S., Carbonell, J., Tomita, M., and Goodman, K. (1992). *Machine Translation: A Knowledge-Based Approach.* Morgan Kaufmann, San Mateo, CA.

Nirenburg, S. and Nirenburg, I. (1988). A framework for lexical selection in natural language generation. In *Proceedings of Twelveth International Conference on Computational Linguistics*, pages 471–475, Budapest, Hungary.

Olsen, M. (1994). *A Semantic and Pragmatic Model of Lexical and Grammatical Aspect.* PhD thesis, Northwestern University, Evanston, IL.

Pesetsky, D. (1982). *Paths and Categories.* PhD thesis, MIT, Cambridge, MA.

Pinker, S. (1989). *Learnability and Cognition: The Acquisition of Argument Structure.* MIT Press, Cambridge, MA.

Proctor, P. (1978). *Longman Dictionary of Contemporary English.* Longman, London, England.

Quantz, J. and Schmitz, B. (1994). Knowledge-based disambiguation for machine translation. *International Journal of Minds and Machines, Special Issue on Knowledge Representation for Natural Language*, 4:39–57.

Sanfilippo, A. and Poznanski, V. (1992). The acquisition of lexical knowledge from combined machine-readable dictionary resources. In *Proceedings of the Applied Natural Language Processing Conference*, pages 80–87, Trento, Italy.

Schabes, Y. (1990). *Mathematical and Computational Aspects of Lexicalized Grammars.* PhD thesis, Computer Science Department, University of Pennsylvania, Philadelphia, PA.

Schubert, L. (1991). Semantic nets are in the eye of the beholder. In Sowa, J., editor, *Principles of Semantic Networks*, pages 95–107. Morgan Kaufmann, San Mateo, CA.

Selman, B. and Hirst, G. (1985). A rule-based connectionist parsing system. In *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, pages 212–219.

Seneff, S. (1992). Tina: A natural language system for spoken language applications. *Computational Linguistics*, 18(1):61–86.

Shapiro, A. (1993). Natural language processing using a propositional semantic network with structured variables. *International Journal of Minds and Machines, Special Issue on Knowledge Representation for Natural Language*, 3:421–451.

Small, S. (1981). *Word Expert Parsing: A Theory of Distributed Word-Based Natural Language Understanding.* PhD thesis, University of Maryland, College Park, MD.

Sondheimer, N., Cumming, S., and Albano, R. (1990). How to realize a concept: Lexical selection and the conceptual network in text generation. *Machine Translation*, 5(1):57–78.

Sowa, J. (1991). Toward the expressive power of natural language. In Sowa, J., editor, *Principles of Semantic Networks*, pages 157–189. Morgan Kaufmann, San Mateo, CA.

Stede, M. (1993). Lexical options in multilingual generation from a knowledge base. Technical Report Manuscript, University of Toronto, Toronto, Canada.

Stevenson, S. (1994). *A Competitive Attachment Model for Resolving Syntactic Ambiguities in Natural Language Parsing.* PhD thesis, University of Maryland, College Park, MD.

Suh, S. (1993). How to process constituent structure in head final languages: The case of korean. In *Proceedings of Chicago Linguistic Society, No. 29.*

Suh, S. (1994). *Constituent Structure Processing and Garden Path Phenomena in Korean.* PhD thesis, University of Maryland, College Park, MD.

Talmy, L. (1978). Semantics and syntax of motion. In Kimball, J., editor, *Syntax and Semantics. Vol. 4, Tense and Aspect*, pages 181–238. Academic Press, New York.

Talmy, L. (1991). Path to realization: A typology of event conflation. Technical Report Buffalo Papers in Linguistics. Vol. 91-01:147-187, State University of New York, Buffalo, NY.

Templeton, M. and Burger, J. (1983). Problems in natural language interface to dbms with examples for eufid. In *Proceedings of the Conference on Applied Natural Language Processing*, pages 3–16.

Thompson, B. and Thompson, F. (1983). Introducing ask, a simple knowledgeable system. In *Proceedings of the Conference on Applied Natural Language Processing*, pages 17–24.

Tomita, M. (1986). *Efficient Parsing for Natural Language.* Kluwer Academic Publishers, Norwell, Massachusetts.

Tsujii, J. and Ananiadou, S. (1993). Knowledge-based processing in mt. In *In Manuscript, U.S. - Japan Machine-Aided Translation Workshop, Washington DC.*

van Riemsdijk, H. and Williams, E. (1986). *Introduction to the Theory of Grammar. Current Studies in Linguistics.* The MIT Press, Cambridge, Massachusetts.

Vijay-Shanker, K. (1987). *A Study of Tree Adjoining Grammars.* PhD thesis, Department of Computer and Information Science, University of Pennsylvania.

Vijay-Shanker, K. (1992). Using descriptions of trees in a tree adjoining grammar. *Computational Linguistics*, 18(4):481–517.

Waltz, D. and Pollack, J. (1985). Massively parallel parsing: A strongly interactive model of natural language interpretation. *Cognitive Science*, 9:51–74.

Whitelock, P. (1992). Shake-and-bake translation. In *Proceedings of Fourteenth International Conference on Computational Linguistics*, pages 784–791, Nantes, France.

Wilks, Y., Fass, D., Guo, C., McDonald, J., and Plate, T. (1990). Providing machine tractable dictionary tools. *Machine Translation*, 5(2):99–154.

Wilks, Y., Fass, D., Guo, C., McDonald, J., Plate, T., and B.Slator (1989). A tractable machine dictionary as a resource for computational semantics. In Boguraev, B. and Briscoe, T., editors, *Computational Lexicography for Natural Language Processing*, pages 193–228. Longman, London.

Woods, W. and Brachman, R. (1978). Research in natural language understanding. Technical Report Quarterly Technical Report Progress Report No. 1, Bolt, Beranek, and Newman, Cambridge, MA.

Wu, Z. and Palmer, M. (1994). Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico.

XTAG (1995). A lexicalized tree adjoining grammar for english. Technical Report The XTAG Research Group, University of Pennsylvania, Philadelphia, PA.

A Hangul Conversion Program

The Hangul Conversion program can convert type of Korean code (Hangul). The supported types are KSC-5601 complete code, Trigem combination code, Yale romanization code, and user definable romanization code.

```
Usage: hconv [Options] [Input_File [Output_File]]

Options:
        -h, --help                  display this help and exit
        -k, --ksc                   KSC code
        -r, --roman                 Roman code
        -t, --trigem, --combo       Trigem code, Combination code
        -T, --table FILENAME        Roman table name
        --rb, --roman-begin CHAR    Roman begin code
        --rd, --roman-div CHAR
        --rs, --roman-sep CHAR      Roman divider code
        --re, --roman-end CHAR      Roman end code
        --ud, --use-divider CHAR    Use divider in roman output
        --cap, --capitalize         Capitalize code in roman output
        -i, --input FILENAME        input filename
        -o, --output FILENAME       output filename
        -V, --version               output version information and exit

Roman code table:
* {ChoSeng} must be in uppercases.
* {CwungSeng} and {CongSeng} must be in lowercases.
* All of Yale keys must be defined in the table.
* Other keys are can be omitted if Yale key == Other key.
* Look into hconv.tab as an example

Default Values:
* input code = ksc
* output code = roman
* roman table file name = <NONE> ‘hconv.tab’ is provided
                                  as an example
* roman begin char = {
* roman divider    = .
* roman end char   = }
* use roman divider in output = NO
* use capitalized code in roman output = YES
```

B BNF for Syntactic Output of Parser

```
Tree     :== Nodes
Nodes    :== (Node Nodes)
Node     :== (Name Options) (Attrs)
Name     :==  CP | Cbar | C | IP | Ibar | I | VP | Vbar | V |
              NP | Nbar | N | AP | Abar | A | PP | Pbar | P |
              XP                          ; This is for spec CP
              Adjunct |                   ; This is for adjunct
```

```
             CONJ |                     ; This is for conjunction
             DISJ |                     ; This is for disjunction
             PRO  |                     ; This is for pro
             Trace |                    ; This is for trace (base position)
             Det | Be | Aux
Options :== Number | Literal | Number Literal | e
Number  :== [0-9]+
Literal :== "[A-Za-z]+"                 ; This is for input literal
Attrs   :== Attr | Attr Attrs | e
Attr    :== [+-]BinAttr | (EnumAttr)
BinAttr :==
        adv             ;adverbial
        adv_s           ;Adverbs modifying sentences
        adv_v           ;Adverbs modifying sentences
        aux             ;auxilary verb
        att             ;attributive
        be              ;a form of be
        bare_inf        ;infinitive
        ca              ;a case assigner
        cap             ;should be capitalized
        cm              ;case-marking indicator
        cmp             ;has comparative form
        ct              ;countable (noun)
        det             ;has determiner
        easy            ;like the word 'easy'
        event           ;NPs that are events
        free_rel        ;relative clause
        gap             ;has a gap
        genitive        ;pronoun
        govern          ;head is a governor
        have            ;a form of have
        head_final
        hyp             ;be hyphen appended to other words
        infl            ;must have overt inflection
        inv             ;inverted auxiliary verb
        land_np         ;landing site of np-movement
        land_wh         ;landing site of wh-movement
        last_conj       ;and/or are +last_conj, either/both are -last_conj
        loc             ;locational PP
        neg             ;negation
        npbarrier       ;crossed a blocking link
        nppg            ;proper government
        nptrace         ;contains an np-trace
        passive         ;passive voice
        perf            ;perfective aspect
        plu             ;plural
        pn              ;proper noun
        ppro            ;PRO is protected
        prd             ;predicative
        pro             ;PRO subject
```

```
        prog            ;progressive aspect
        pron            ;pronoun
        refl            ;reflective pronoun
        reg             ;regular suffixing stem+s,+ing,+ed,+er,+est
        restrictive     ;restrictive relative clause
        superlative     ;form of adjective
        theta           ;assign external theta-role or need theta-role
        theta_assigner  ;assign theta-role to complement
        topic           ;no need to say it's wh
        wh              ;wh-element
        whbarrier       ;crossed a blocking link
        whpg            ;proper government
        whtrace         ;contains a wh-trace
        whvacuous                       ;vacuous movement


ListAttrs :==
        aform [norm|er|est]             ;the type of adjective
        auxform                         ;the form of auxiliary verb
          [to|can|could|do|did|does|may|might|would|should|
           must|will|ought|shall|have_to|be_going_to]+
        case [acc|nom]                  ;the case of an NP
        cform                           ;the type of clauses
          [fin|inf|npsc|apsc|ppsc|vpsc]
        comp                            ;the type of complementizers
          [for|that|whether|if|other|none]
        nform [there|it|norm|ing]       ;the type of NP
        per [1|2|3]                     ;person
        pform                           ;the preposition form
          [aboard|about|above|according_to|across|afore|after|against|
           agin|along|alongside|amid|amidst|among|amongst|anent|around|
           as|aslant|astride|at|athwart|bar|before|behind|below|beneath|
           beside|besides|between|betwixt|beyond|but|by|circa|despite|
           down|during|ere|except|for|from|in|inside|into|less|like|mid|
           midst|minus|near|next|nigh|nigher|nighest|notwithstanding|of|
           off|on|on_to|onto|out|outside|over|past|pending|per|plus|qua|
           re|round|sans|save|since|through|throughout|thru|till|to|
           toward|towards|under|underneath|unlike|until|unto|up|upon|
           versus|via|vice|with|within|without]
        pred [n|v|a|p]                  ;type of predicate
        vform [bare|s|ed|ing]           ;inflection of verb
        rare [very|very_very]
        tense                           ;tense
          [present|past|future|pastfut]
        whform                          ;form of wh-element
          [what|where|who|whose|which|when|why|whom|whoso|whosever|how|
           whatever|whichever|whoever|whichsoever|whosoever|whatsoever|
           whomever|whomsoever]
```

C Korean Lexical Entries for Parser and LCS Composition

C.1 Parser Entries

```
(Sara (meaning main (np)))

(John (meaning main (np)))

(Bill (meaning main (np)))

; Object NP
; this ends with -ul or -lul

(towum-ul (meaning main (obj_np)))

(Bill-ul (meaning main (obj_np)))

; Nominative NP
; this ends with -i, -un, or -kk

(John-i (meaning main (np (case nom))))

(Bill-un (meaning main (np (case nom))))

; Korean P
; need to be realized as a separate word.
; In Korean, this is a marker, which attached to the end of N
; such as John-eykey, Sally-wa.
; Morphological analyzer has to separate ‘Korean P’ from the root.
;   John-eykey ---> John eykey
;   Sally-wa   ---> Sally wa

(eykey   (meaning main (pp (pform to))   (((case acc) (cat n))) (node P)))

(wa      (meaning main (pp (pform with)) (((case acc) (cat n))) (node P)))

(eykeyse (meaning main (pp (pform from)) (((case acc) (cat n))) (node P)))

; Korean V
; Since I in Korean is of infix form in verb, this will give Kimmo
;  hard time.

(cwuessta (meaning main (verb past-tense -passive)
                   ([((cat p) (pform to)) (obj_np)]))
          (meaning main (verb) ((obj_np))))

(patassta (meaning main (verb past-tense -passive)
                   ([((cat p) (pform from)) (obj_np)]))
          (meaning main (verb) ((obj_np))))

(kyelhonhayssta (meaning main (verb past-tense -passive) ((pp (pform with)))))
```

```
;
; English ones
; The lexical entries used in generating parse tree.

(helped (meaning main (verb past-tense -passive) ((obj_np))))
(married (meaning main (verb past-tense -passive) ((obj_np))))
```

C.2 LCS Entries

```
;; Root form: cwu-(ta)
(
 :DEF_WORD "cwuessta"
 :COMMENT "give: John gave help to Bill"
 :LANGUAGE korean
 :LCS (cause (* thing 1)
       (go poss (* thing 2)
        ((* to 5) poss (thing 2) (at poss (thing 2) (thing 6)))
        (from poss (thing 2) (at poss (thing 2) (thing 1))))
       (givingly 25))
 :VAR_SPEC ((1 (UC (animate +))) (25 :conflated))
)

;; Root form: pat-(ta)
(
 :DEF_WORD "patassta"
 :COMMENT "receive: Bill received help from John"
 :LANGUAGE korean
 :LCS (let (* thing 1)
        (go poss (* thing 2)
            (to poss (thing 2) (at poss (thing 2) (thing 1)))
            ((* from 3) poss (thing 2) (at poss (thing 2) (thing 4))))
        (receivingly 25))
 :VAR_SPEC ((1 (UC (animate +))) (3 :optional) (25 :conflated))
)

;; Root form: top-(ta)
(
 :DEF_WORD "towassta"
 :COMMENT "help: John helped Bill"
 :LANGUAGE korean
 :LCS (cause (* thing 1)
       (go poss (help 2)
        (to poss (help 2) (at poss (help 2) (* thing 6)))
        (from poss (help 2) (at poss (help 2) (thing 1)))))
 :VAR_SPEC ((1 (UC (animate +))) (2 :conflated)
            (5 (cnd nil 6)) (6 (cnd nil 6)))
)

;; Root form: kyelhonha-(ta)
(
```

```
:DEF_WORD "kyelhonhayssta"
:COMMENT "marry: John married with Sally"
:LANGUAGE korean
:LCS (go loc (* thing 2)
         (toward loc (thing 2) ((* co 9) loc (thing 2) (thing 6)))
         (marryingly 25))
:VAR_SPEC ((2 (UC (human +))) (6 (uc (human +))) (25 :conflated))
)

(
 :DEF_WORD "eykey"
 :COMMENT "to"
 :LANGUAGE korean
 :LCS (to poss (thing 2) (at poss (thing 2) (* thing 6)))
)


(
 :DEF_WORD "eykeyse"
 :COMMENT "from"
 :LANGUAGE korean
 :LCS (from poss (thing 2) (at poss (thing 2) (* thing 1)))
)

(
 :DEF_WORD "wa"
 :COMMENT "with"
 :LANGUAGE korean
 :LCS (co loc (thing 2) (* thing 6))
)

;; Root form: John
(
 :DEF_WORD "John"
 :LANGUAGE korean
 :LCS (john 0)
 :VAR_SPEC ((0 (UC (human +)))))

;; Root form: Sally
(
 :DEF_WORD "Sally"
 :LANGUAGE korean
 :LCS (sally 0)
 :VAR_SPEC ((0 (UC (human +)))))

;; Root form: Sara
(
 :DEF_WORD "Sara"
 :LANGUAGE korean
 :LCS (sara 0)
```

```
 :VAR_SPEC ((0 (UC (human +)))))

;; Root form: Bill
(
 :DEF_WORD "Bill"
 :LANGUAGE korean
 :LCS (bill 0)
 :VAR_SPEC ((0 (UC (human +)))))


;; Root form: John
(
 :DEF_WORD "John-ul"
 :LANGUAGE korean
 :LCS (john 0)
 :VAR_SPEC ((0 (UC (human +)))))

;; Root form: Sally
(
 :DEF_WORD "Sally-ul"
 :LANGUAGE korean
 :LCS (sally 0)
 :VAR_SPEC ((0 (UC (human +)))))

;; Root form: Sara
(
 :DEF_WORD "Sara-ul"
 :LANGUAGE korean
 :LCS (sara 0)
 :VAR_SPEC ((0 (UC (human +)))))

;; Root form: Bill
(
 :DEF_WORD "Bill-ul"
 :LANGUAGE korean
 :LCS (bill 0)
 :VAR_SPEC ((0 (UC (human +)))))

;; Root form: John
(
 :DEF_WORD "John-i"
 :LANGUAGE korean
 :LCS (john 0)
 :VAR_SPEC ((0 (UC (human +)))))

;; Root form: Sally
(
 :DEF_WORD "Sally-i"
 :LANGUAGE korean
 :LCS (sally 0)
```

```
 :VAR_SPEC ((0 (UC (human +)))))

;; Root form: Sara
(
 :DEF_WORD "Sara-i"
 :LANGUAGE korean
 :LCS (sara 0)
 :VAR_SPEC ((0 (UC (human +)))))

;; Root form: Bill
(
 :DEF_WORD "Bill-i"
 :LANGUAGE korean
 :LCS (bill 0)
 :VAR_SPEC ((0 (UC (human +)))))


;; Root form: John
(
 :DEF_WORD "John-un"
 :LANGUAGE korean
 :LCS (john 0)
 :VAR_SPEC ((0 (UC (human +)))))

;; Root form: Sally
(
 :DEF_WORD "Sally-un"
 :LANGUAGE korean
 :LCS (sally 0)
 :VAR_SPEC ((0 (UC (human +)))))

;; Root form: Sara
(
 :DEF_WORD "Sara-un"
 :LANGUAGE korean
 :LCS (sara 0)
 :VAR_SPEC ((0 (UC (human +)))))

;; Root form: Bill
(
 :DEF_WORD "Bill-un"
 :LANGUAGE korean
 :LCS (bill 0)
 :VAR_SPEC ((0 (UC (human +)))))

;; Root form: towum
(
 :DEF_WORD "towum-ul"
 :COMMENT "help"
 :LANGUAGE korean
```

```
 :LCS (help 0)
 :VAR_SPEC ((0 (UC (animte -)))))
```

D Parser and LCS Composition Output

D.1 E: John married Sally

Syntactic Parse:

```
((CP)(-inv)
 ((Cbar)(-inv)
    ((IP)(+ca)
      ((NP)(np -adv)
        ((Nbar)(np -adv)
          ((N "John")(np pnp -adv))))
      ((Ibar)(+govern)
        ((VP)(+ca)
          ((Vbar)(+ca)
            ((V "married")(verb +ca))
            ((NP)(np -adv)
              ((Nbar)(np -adv)
                ((N "Sally")(np -adv)))))))))
```

Composed LCS:

```
(:ROOT EVENT GO LOCATIONAL NIL 1
 ((:SUB THING JOHN NIL NIL 2)
  (:ARG PATH TOWARD LOCATIONAL NIL 3
   ((:SUB THING JOHN NIL NIL 2)
    (:ARG POSITION CO LOCATIONAL NIL 4
     ((:SUB THING JOHN NIL NIL 2) (:ARG THING SALLY NIL NIL 5)))))
  (:MOD MANNER MARRYINGLY NIL NIL 6)))
```

D.2 E: John helped Bill

Syntactic Parse:

```
((CP)(-inv)
  ((Cbar)(-inv)
    ((IP)(+ca)
      ((NP)(np -adv)
        ((Nbar)(np -adv)
          ((N "John")(np pnp -adv))))
      ((Ibar)(+govern)
        ((VP)(+ca)
          ((Vbar)(+ca)
            ((V "helped")(verb +ca))
            ((NP)(np -adv)
              ((Nbar)(np -adv)
                ((N "Bill")(np -adv)))))))))
```

Composed LCS:

```
(:ROOT EVENT CAUSE NIL NIL 1
 ((:SUB THING JOHN NIL NIL 2)
  (:ARG EVENT GO POSSESSIONAL NIL 3
   ((:SUB THING HELP NIL NIL 4)
    (:ARG PATH FROM POSSESSIONAL NIL 5
     ((:SUB THING HELP NIL NIL 4)
      (:ARG POSITION AT POSSESSIONAL NIL 6
       ((:SUB THING HELP NIL NIL 4) (:SUB THING JOHN NIL NIL 2)))))
    (:ARG PATH TO POSSESSIONAL NIL 7
     ((:SUB THING HELP NIL NIL 4)
      (:ARG POSITION AT POSSESSIONAL NIL 8
       ((:SUB THING HELP NIL NIL 4) (:ARG THING BILL NIL NIL 9)))))))))
```

D.3 K: John-i Sally-wa kyelhonhayssta (John married (with) Sally)

Syntactic Parse:

```
((CP)(-inv)
  ((Cbar)(-inv)
    ((IP)(+ca)
      ((NP)(np -adv)
        ((Nbar)(np -adv)
          ((N "John-i")(np pnp -adv))))
      ((Ibar)(+govern)
        ((VP)(+ca)
          ((Vbar)(+ca)
            ((PP)(+theta_assigner)
              ((Pbar)(+ca)
                ((NP)(np -adv)
                  ((Nbar)(np -adv)
                    ((N "Sally")(np -adv))))
                ((P "wa")(+ca))))
            ((V "kyelhonhayssta")(past-tense verb +ca)))))))
```

Composed LCS:

```
(:ROOT EVENT GO LOCATIONAL NIL 1
 ((:SUB THING JOHN NIL NIL 2)
  (:ARG PATH TOWARD LOCATIONAL NIL 3
   ((:SUB THING JOHN NIL NIL 2)
    (:ARG POSITION CO LOCATIONAL NIL 4
     ((:SUB THING JOHN NIL NIL 2) (:ARG THING SALLY NIL NIL 5)))))
  (:MOD MANNER MARRYINGLY NIL NIL 6)))
```

D.4 K: Bill-un John eykeyse towum-ul patassta (Bill received help from John)

Syntactic Parse:

```
((CP)(-inv)
  ((Cbar)(-inv)
    ((IP)(+ca)
      ((NP)(np -adv)
        ((Nbar)(np -adv)
```

```
          ((N "Bill-un")(np pnp -adv))))
      ((Ibar)(+govern)
        ((VP)(+ca)
          ((Vbar)(+ca)
            ((PP)(+theta_assigner)
              ((Pbar)(+ca)
                ((NP)(np -adv)
                  ((Nbar)(np -adv)
                    ((N "John")(np pnp -adv))))
                ((P "eykeyse")(+ca))))
            ((NP)(np -adv)
              ((Nbar)(np -adv)
                ((N "towum-ul")(np -adv))))
            ((V "patassta")(past-tense verb +ca)))))))
```

Composed LCS:

```
(:ROOT EVENT LET NIL NIL 1
 ((:SUB THING BILL NIL NIL 2)
  (:ARG EVENT GO POSSESSIONAL NIL 3
   ((:SUB THING HELP NIL NIL 4)
    (:ARG PATH FROM POSSESSIONAL NIL 5
     ((:SUB THING HELP NIL NIL 4)
      (:ARG POSITION AT POSSESSIONAL NIL 6
       ((:SUB THING HELP NIL NIL 4) (:ARG THING JOHN NIL NIL 7)))))
    (:ARG PATH TO POSSESSIONAL NIL 8
     ((:SUB THING HELP NIL NIL 4)
      (:ARG POSITION AT POSSESSIONAL NIL 9
       ((:SUB THING HELP NIL NIL 4) (:SUB THING BILL NIL NIL 2)))))))
  (:MOD MANNER RECEIVINGLY NIL NIL 10)))
```

D.5 K: John-i Bill-eykey towum-ul cwuessta. (John gave help to Bill)

Syntactic Parse:

```
((CP)(-inv)
  ((Cbar)(-inv)
    ((IP)(+ca)
      ((NP)(np -adv)
        ((Nbar)(np -adv)
          ((N "John-i")(np pnp -adv))))
      ((Ibar)(+govern)
        ((VP)(+ca)
          ((Vbar)(+ca)
            ((PP)(+theta_assigner)
              ((Pbar)(+ca)
                ((NP)(np obj_np -cm)
                  ((Nbar)(np obj_np -ca)
                    ((N "Bill")(np obj_np -ca))))
                ((P "eykey")(+ca))))
            ((NP)(np -adv)
              ((Nbar)(np -adv)
```

```
                ((N "towum-ul")(np -adv))))
            ((V "cwuessta")(past-tense verb +ca))))))))
```

Composed LCS:

```
(:ROOT EVENT CAUSE NIL NIL 1
 ((:SUB THING JOHN NIL NIL 2)
  (:ARG EVENT GO POSSESSIONAL NIL 3
   ((:SUB THING HELP NIL NIL 4)
    (:ARG PATH FROM POSSESSIONAL NIL 5
     ((:SUB THING HELP NIL NIL 4)
      (:ARG POSITION AT POSSESSIONAL NIL 6
       ((:SUB THING HELP NIL NIL 4) (:SUB THING JOHN NIL NIL 2)))))
    (:ARG PATH TO POSSESSIONAL NIL 7
     ((:SUB THING HELP NIL NIL 4)
      (:ARG POSITION AT POSSESSIONAL NIL 8
       ((:SUB THING HELP NIL NIL 4) (:ARG THING BILL NIL NIL 9)))))))
  (:MOD MANNER GIVINGLY NIL NIL 10)))
```

D.6 K: John-i Bill-ul towassta (John helped Bill)

Syntactic Parse:

```
((CP)(-inv)
  ((Cbar)(-inv)
    ((IP)(+ca)
      ((NP)(np -adv)
        ((Nbar)(np -adv)
          ((N "John-i")(np pnp -adv))))
      ((Ibar)(+govern)
        ((VP)(+ca)
          ((Vbar)(+ca)
            ((NP)(np -adv)
              ((Nbar)(np -adv)
                ((N "Bill")(np -adv))))
            ((V "towassta")(past-tense verb +ca))))))))
```

Composed LCS:

```
(:ROOT EVENT CAUSE NIL NIL 1
 ((:SUB THING JOHN NIL NIL 2)
  (:ARG EVENT GO POSSESSIONAL NIL 3
   ((:SUB THING HELP NIL NIL 4)
    (:ARG PATH FROM POSSESSIONAL NIL 5
     ((:SUB THING HELP NIL NIL 4)
      (:ARG POSITION AT POSSESSIONAL NIL 6
       ((:SUB THING HELP NIL NIL 4) (:SUB THING JOHN NIL NIL 2)))))
    (:ARG PATH TO POSSESSIONAL NIL 7
     ((:SUB THING HELP NIL NIL 4)
      (:ARG POSITION AT POSSESSIONAL NIL 8
       ((:SUB THING HELP NIL NIL 4) (:ARG THING BILL NIL NIL 9))))))))))
```

E Korean Morphology Dictionary

We received a hardcopy of a Korean lexicon from the Army Research Institute, for which there was no online backup. This dictionary was scanned by OCR and then proofread and corrected manually during spring of 1994. There are about 1300 words. We are in the process of converting this dictionary into a morphology dictionary. Mr. Lee has inserted dots for syllabification into the first 18% of the Korean words. Our goal is to complete this syllabification task and also to "import" LCS's and theta-roles using automatic techniques such as those described in section 7.

```
WORD          $ POS         $ TRANSLATE            $ CASE
gun.in        $ n           $ soldier              $ N
ga.ge         $ n           $ store                $ N
gun.dae       $ n           $ military             $ N
gae           $ n           $ dog                  $ N
na            $ n           $ I                    $ N
ga.gu         $ n           $ furniture            $ N
ga.ro.su      $ n           $ roadside trees       $ N
ga.bang       $ n           $ bag                  $ N
ga.wi         $ n           $ scissors             $ N
ga.jog        $ n           $ family               $ N
gan.jang      $ n           $ soy sauce            $ N
gan.pan       $ n           $ sign board           $ N
gal.bi        $ n           $ rib                  $ N
gam.gi        $ n           $ cold                 $ N
gang          $ n           $ river                $ N
gang.dang     $ n           $ auditorium           $ N
ga.gu.jeom    $ n           $ furniture store      $ N
ga.neung.seong $ n          $ possibility          $ N
ga.seum       $ n           $ chest, breast        $ N
ga.eul        $ n           $ autumn, fall         $ N
ga.jeong.bu   $ n           $ maid                 $ N
gan.cheob     $ n           $ spy, infiltrator     $ N
ga.nho.weon   $ n           $ nurse                $ N
gang.eui      $ n           $ lecture              $ N
gae.ul        $ n           $ brook                $ N
gae.weol      $ n           $                      $ N
gae.chal.gu   $ n           $ ticket gate          $ N
geo.ri        $ n           $ street               $ N
geo.sil       $ n           $ living room          $ N
geo.ul        $ n           $ mirror               $ N
geon.mul      $ n           $ building             $ N
geom.sa       $ n           $ inspection           $ N
geom.yeol     $ n           $ inspection           $ N
ge.yang.dae   $ n           $ pole                 $ N
gyeo.ul       $ n           $ winter               $ N
gyeol.jeong   $ n           $ decision             $ N
gyeol.hon     $ n           $ marriage             $ N
gyeong.bi     $ n           $ guard                $ N
gyeong.u      $ n           $ occasion             $ N
gyeong.je     $ n           $ economy              $ N
gyeong.chal   $ n           $ police               $ N
gyeong.chi    $ n           $ view, scenery        $ N
gyeong.heom   $ n           $ experience           $ N
gye.geub      $ n           $ rank                 $ N
gye.dan       $ n           $ stairs               $ N
gye.ran       $ n           $ egg                  $ N
gye.san.seo   $ n           $ check                $ N
gye.sog       $ n           $ continuance          $ N
gye.hoeg      $ n           $ plan                 $ N
go.mo         $ n           $ paternal aunt        $ N
gosog         $ n           $ high speed           $ N
go.jang       $ n           $ malfunction          $ N
go.chu        $ n           $ red pepper           $ N
go.cheung     $ n           $ highrise             $ N
go.hyang      $ n           $ hometown             $ N
gol.mog       $ n           $ alley                $ N
gos           $ n           $ place                $ N
gong          $ n           $ ball                 $ N
gong.gyeog    $ n           $ attack               $ N
gong.gun      $ n           $ air force            $ N
gong.gi       $ n           $ air                  $ N
gong.mun      $ n           $ official letter      $ N
gong.bu       $ n           $ study                $ N
gong.sa       $ n           $ construction         $ N
gong.san.dang $ n           $ communist party      $ N
gong.sig      $ n           $ formality            $ N
gong.weon     $ n           $ park                 $ N
gong.jang     $ n           $ factory              $ N
gong.chaeg    $ n           $ notebook             $ N
gong.hang     $ n           $ airport              $ N
gwa           $ n           $ lesson               $ N
gwa.geo       $ n           $ past                 $ N
gwan.gye      $ n           $ relation             $ N
gwa.il        $ n           $ fruit                $ N
gwa.ja        $ n           $ cookie               $ N
gwan.sim      $ n           $ concern              $ N
gwan.ri       $ n           $ government official  $ N
gwan.je.tab   $ n           $ control tower        $ N
gwang.jang    $ n           $ plaza                $ N
gyo.gwa.seo   $ n           $ textbook             $ N
gyo.gwan      $ n           $ instructor           $ N
```

```
gyo.sil $ n $ classroom $ N
gyo.yug $ n $ education $ N
gyo.tong $ n $ traffic $ N
gyo.hwan $ n $ operator $ N
gu $ n $ nine $ N
gu.nae $ n $ extension $ N
gu.reum $ n $ cloud $ N
gu.seog $ n $ corner $ N
gug $ n $ soup $ N
gug.ga $ n $ country $ N
gug.gyeong $ n $ national boundary $ N
gug.gun $ n $ ROK armed forces $ N
gug.gi $ n $ national flag $ N
gug.nae.seon $ n $ domestic lines $ N
gug.su $ n $ noodle $ N
gun $ n $ army $ N
gun.dan $ n $ corps $ N
gun.sa.ryeog $ n $ military might $ N
 $ $ military force $
gweon.chong $ n $ pistol $ N
gwi $ n $ ear $ N
gyul $ n $ tangerine $ N
geu $ n $ he $ N
geu.rim $ n $ painting $ N
geug.jang $ n $ theater $ N
geun.cheo $ n $ vicinity $ N
geum.aeg $ n $ an amount of money $ N
geum.yeon $ n $ no smoking $ N
geum.yo.il $ n $ Friday $ N
geub.haeng $ n $ express $ N
gi.gan $ n $ period $ N
gi.gye $ n $ machine $ N
gi.gwan.chong $ n $ machinegun $ N
gi.rog $ n $ record $ N
gi.reum $ n $ oil $ N
gi.bun $ n $ feeling, mood $ N
gi.sa $ n $ article $ N
gi.sug.sa $ n $ dormitory $ N
gi.sul.ja $ n $ technician $ N
gi.on $ n $ temperature $ N
gi.ji $ n $ base, camp $ N
gi.cha $ n $ train $ N
gi.ta $ n $ the others $ N
gi.hoe $ n $ chance $ N
gi.hu $ n $ climate $ N
gil $ n $ road, way $ N
gim.chi $ n $ pickled cabbage $ N
ggag.du.gi $ n $ hot pickled cabbage $ N
 $ $ hot pickled radish $
ggang.tong $ n $ can $ N
ggo.ma $ n $ kid, a little one $ N
ggog.dae.gi $ n $ top, summit $ N
ga $ v $ go $ N
ga.ggab $ v $ be near $ N
ga.byeob $ v $ be light $ N
ga.ji $ v $ take into possession $ N
gan.dan.ha $ v $ be simple $ N
gal $ v $ replace $ N
gam.sa.ha $ v $ thank $ N
gat $ v $ be the same $ N
geo.seu.reu $ v $ change money into smaller $ N
geon.gang.ha $ v $ be healthy $ N
geon.neo $ v $ cross $ N
geol $ v $ hang $ N
geolri $ v $ be hang $ N
geolri $ v $ take (time) $ N
geod $ v $ walk $ N
ge.eu.reu $ v $ be lazy $ N
gyeol.geun.ha $ v $ be absent from work $ N
gyeol.seog.ha $ v $ be absent from school $ N
gyeol.jeong.ha $ v $ decide $ N
gyeol.hon.ha $ v $ marry $ N
go.dan.ha $ v $ be tired $ N
go.mab $ v $ thank $ N
go.chi $ v $ repair, fix $ N
go.peu $ v $ be hungry $ N
gwaen.chanh $ v $ be alright $ N
goeng.jang.ha $ v $ be extreme $ N *
gu.gyeong.ha $ v $ look at $ N
gu.byeol.ha $ v $ distinguish $ N
gu.ha $ v $ seek $ N
gub $ v $ broil $ N
geumandu $ v $ quit $ N
geu.chi $ v $ stop $ N
geun.mu.ha $ v $ work $ N
geub.ha $ v $ urgent $ N
gi.da.ri $ v $ wait $ N
si.jag.ha $ v $ begin $ N
gi.bbeu $ v $ be happy $ N
gi.bbeo.ha $ v $ be happy $ N
gil $ v $ be long $ N
gip $ v $ be deep $ N
gga.da.rob $ v $ be particular, fussy $ N
 $ $ be picky $
gga.mah $ v $ be black, dark $ N
ggagg $ v $ peel $ N
gga.ggeus.ha $ v $ be clean $ N
```

```
ggae $ v $ wake up $ N
ggae $ v $ break $ N
ggoj $ v $ stick in $ N
ggeu $ v $ turn off $ N
ggeunh $ v $ hang up, cut off $ N
ggeulh.i $ v $ boil $ N
ggeulh $ v $ boil $ N
ggeut.na $ v $ end $ N
ggeut.nae $ v $ finish $ N
ggi $ v $ cloud up $ N
ggi.u $ v $ put .. between .. $ N
      $ $ insert .. between .. $
ga.reu.chi $ v $ teach $ N
ga $ p $ $ Y
ga.ggeum $ adv $ occasionally $ N
ga.gga.i $ adv $ near $ N
ga.jang $ adv $ the most $ N
gab.ja.gi $ adv $ suddenly $ N
gat.i $ adv $ together $ N
geo.eui $ adv $ almost $ N
go $ adj $ that very $ N
i $ p $ $ Y
gwa $ p $ and $ Y
wa $ p $ and $ Y
goeng.jang.hi $ adv $ magnificently $ N
geu.nyang $ adv $ as it is $ N
geu.rae.seo $ adv $ so $ N
geu.reo.meu.ro $ adv $ therefore $ N
geu.rae.do $ adv $ still $ N
geu.reo.myeon $ adv $ then $ N
geu.reon.de $ adv $ by the way $ N
geu.reoh.ge $ adv $ like that $ N
geu.geon.e $ adv $ previously $ N
geum.bang $ adv $ immediately $ N
gga.ji $ p $ until $ Y
ggog $ adv $ exactly, surely $ N
ggwae $ adv $ quite $ N
god $ adv $ right away $ N
geu.dong.an $ adv $ during that time $ N
god.jang $ adv $ straight ahead $ N
geu.ddae $ adv $ that time $ N
geu.dae.ro $ adv $ like that $ N
na.ra $ n $ nation $ N
na.mu $ n $ tree $ N
na.mul $ n $ seasoned vegetable $ N
na.i $ n $ age $ N
na.heul $ n $ four days $ N
nal $ n $ day $ N
nal.ssi $ n $ weather $ N
nal.jja $ n $ date $ N
nam $ n $ south $ N
nam.mae $ n $ brother and sister $ N
nam.ja $ n $ male $ N
nam.pyeon $ n $ husband $ N
naj $ n $ day $ N
nae.nyeon $ n $ next year $ N
nae.mu.ban $ n $ barracks $ N
nae.oe $ n $ couple $ N
nae.yong $ n $ content $ N
nae.il $ n $ tomorrow $ N
neo $ n $ you $ N
ne $ n $ you $ N
ne.geo.ri $ n $ intersection $ N
neg.ta.i $ n $ necktie $ N
nes $ n $ four $ N
nyeon $ n $ year $ N
no.teu $ n $ notebook $ N
no.rae $ n $ song $ N
no.in $ n $ the old $ N
nog.eum.gi $ n $ tape recorder $ N
non $ n $ rice paddy $ N
nong.gu $ n $ basketball $ N
nong.dam $ n $ joke $ N
nong.sa $ n $ farming $ N
nong.eob $ n $ agriculture $ N
nong.jang $ n $ farm $ N
nu.gu $ n $ who, whom $ N
nu.na $ n $ male's older sister $ N
nun $ n $ eye, snow $ N
nyu.seu $ n $ news $ N
neung.ryeog $ n $ ability $ N
nae $ n $ I $ N
na.ga $ v $ go out $ N
na $ v $ happen, be produced $ N
na.reu $ v $ carry, move $ N
nab.beu $ v $ be bad $ N
na.o $ v $ come out $ N
na.ta.na $ v $ appear $ N
nam $ v $ remain $ N
nas $ v $ be better, recover $ N
naj $ v $ be low $ N
nae $ v $ pay, submit $ N
nae.ri $ v $ get off $ N
neog.neog.ha $ v $ be enough $ N
neolb $ v $ be wide, spacious $ N
neom $ v $ cross over $ N
neoh $ v $ put in $ N
```

```
no.rah $ v $ be yellow $ N
nol $ v $ play, be out of work $ N
nol.ra $ v $ be surprised $ N
nol.ri $ v $ make fun of $ N
nop $ v $ be high $ N
noh $ v $ put, place $ N
nu.reu $ v $ press $ N
nub $ v $ lie down $ N
neul $ v $ increase $ N
neul.ri $ v $ increase $ N
neulg $ v $ get old $ N
neuj $ v $ be late $ N
nol.ri $ v $ make..play $ N
$ $ make fun of $
na.jung.e $ adv $ later $ N
nae.e $ adv $ within $ N
neo.mu $ adv $ too, excessively $ N
ne $ adj $ four $ N
neun $ p $ $ Y
eun $ p $ $ Y
neul $ adv $ always $ N
da.ri $ n $ bridge, leg $ N
da.bang $ n $ tea house $ N
da.seos $ n $ five $ N
daeum $ n $ next $ N
dalg $ n $ chicken $ N
dambae $ n $ cigarette $ N
damyo $ n $ blanket $ N
dassae $ n $ five days $ N
daegisil $ n $ waiting room $ N
daedae $ n $ battalion $ N
daeryeong $ n $ colonel $ N
daerijeom $ n $ agency $ N
daewi $ n $ captain $ N
daejang $ n $ general $ N
daetongryeong $ n $ president (of a country)$ N
daepyo $ n $ representative $ N
daehanmingug $ n $ Republic of Korea $ N
daehanhanggong $ n $ Korean Airlines $ N
daehabsil $ n $ waiting room $ N
daehoe $ n $ contest $ N
deowi $ n $ heat $ N
Deogsugung $ n $ name of a palace $ N
de $ n $ place $ N
do $ n $ degree $ N
doro $ n $ road $ N
doseogwan $ n $ library $ N
dosi $ n $ city $ N
doggam $ n $ flu $ N
dogbon $ n $ reading book $ N
dogil $ n $ Germany $ N
don $ n $ money $ N
dong $ n $ east $ N
dongne $ n $ village $ N
dongryo $ n $ colleague $ N
dongsamuso $ n $ town office $ N
dongsaeng $ n $ younger brother/sister$ N
dongjeon $ n $ coin $ N
doeji $ n $ pig $ N
dul $ n $ two $ N
duljjae $ n $ second $ N
dwi $ n $ rear, back $ N
deung $ n $ lamp $ N
deung $ n $ etc. $ N
ddal $ n $ daughter $ N
ddang $ n $ earth, ground $ N
ddaemun $ n $ cause $ N
ddae $ n $ moment, time $ N
ddeog $ n $ rice cake $ N
dduggeong $ n $ lid, cover $ N
ddeus $ n $ meaning $ N
rosdehotel $ n $ Hotel Lotte $ N
riteo $ n $ liter $ N
dae $ n $ $ N
da $ adv $ all $ N
dasi $ adv $ again $ N
daegae $ adv $ mostly $ N
daeryag $ adv $ roughly $ N
daechero $ adv $ generally $ N
deo $ adv $ more $ N
deol $ adv $ less $ N
du $ adj $ of two $ N
ddaraseo $ adv $ accordingly $ N
ddaro $ adv $ separately, apart $ N
ddan $ adj $ other $ N
ddoneun $ adv $ or $ N
ddohan $ adv $ as well $ N
rang $ p $ with $ N
irang $ p $ with $ N
ro $ p $ to $ Y
reul $ p $ $ Y
eul $ p $ $ Y
do $ p $ also, too $ N
danl $ v $ attend, go to $ N
dareu $ v $ be different $ N
dachi $ v $ get injured $ N
dagg $ v $ polish, wash $ N
```

```
dad          $ v          $ close                  $ N
dadhi        $ v          $ be closed              $ N
dal          $ v          $ sweet                  $ N
dali         $ v          $ run                    $ N
dalh         $ v          $ wear away, wear down   $ N
danqqi       $ v          $ pull                   $ N
dah          $ v          $ reach, arrive          $ N
daedabha     $ v          $ answer                 $ N
daejeobha    $ v          $ treat                  $ N
daeju        $ v          $ connect (phone), supply$ N
deoreob      $ v          $ be dirty               $ N
deoha        $ v          $ add                    $ N
deob         $ v          $ be hot, warm           $ N
deop         $ v          $ put cover on, close (book)$ N
dochagha     $ v          $ arrive                 $ N
dol          $ v          $ turn, be prevalent     $ N
dolri        $ v          $ turn, dial             $ N
dob          $ v          $ help                   $ N
doe          $ v          $ become                 $ N
deuri        $ v          $ give                   $ N
deud         $ v          $ hear, listen to        $ N
deul         $ v          $ drink, eat             $ N
deul         $ v          $ lift, carry in hand    $ N
deul         $ v          $ cost                   $ N
deul         $ v          $ get in                 $ N
deul         $ v          $ lodge                  $ N
deulreu      $ v          $ stop by                $ N
deulri       $ v          $ be heard               $ N
deulri       $ v          $ be lifted              $ N
deulri       $ v          $ make..lift..           $ N
ddadeusha    $ v          $ be warm                $ N
ddareu       $ v          $ follow                 $ N
ddeona       $ v          $ leave, depart          $ N
ddeoleoji    $ v          $ drop, fall, run out    $ N
ddogddogha   $ v          $ be smart               $ N
ddungddungha $ v          $ be fat                 $ N
ddwi         $ v          $ jump, run              $ N
ddeugeob     $ v          $ be hot                 $ N
ddeu         $ v          $ rise                   $ N
madang       $ n          $ yard                   $ N
masan        $ n          $                        $ N
maeul        $ n          $ village                $ N
mail         $ n          $ mile                   $ N
magnae       $ n          $ the youngest           $ N
magsa        $ n          $ barracks               $ N
man          $ n          $ ten thousand           $ N
mal          $ n          $ words, speech, horse   $ N
malsseum     $ n          $ words, speech          $ N
mas          $ n          $ taste, flavor          $ N
maedal       $ n          $ every month            $ N
maeil        $ n          $ everyday               $ N
maejeom      $ n          $ store, stand           $ N
maepyogu     $ n          $ ticket window          $ N
maepyoso     $ n          $ box(ticket) office     $ N
maegju       $ n          $ beer                   $ N
meori        $ n          $ head, hair             $ N
memoji       $ n          $ memo pad               $ N
myeonjeog    $ n          $ sqaure measure         $ N
myeonhoe     $ n          $ interview              $ N
myeong       $ n          $                        $ N
myeongdan    $ n          $ roster, list of names  $ N
myeongryeong $ n          $ order, command         $ N
myeongseong  $ n          $ fame                   $ N
myeongil     $ n          $ festive day, tomorrow  $ N
myeongjeol   $ n          $ festive season         $ N
myeoch       $ n          $ now many               $ N
myeochmadi   $ n          $ few words              $ N
geos         $ n          $ thing                  $ N
more         $ n          $ the day after tomorrow$ N
moyang       $ n          $ shape, style           $ N
moim         $ n          $ gathering              $ N
moja         $ n          $ hat, cap               $ N
mojib        $ n          $ recruiting             $ N
motungi      $ n          $ corner                 $ N
mogsori      $ n          $ voice                  $ N
mogyoil      $ n          $ Thursday               $ N
mogyog       $ n          $ bathing                $ N
mogjeogji    $ n          $ destination            $ N
mom          $ n          $ body                   $ N
mos          $ n          $ nail                   $ N
mugi         $ n          $ weapon                 $ N
mueos        $ n          $ what, something        $ N
muyeog       $ n          $ trade                  $ N
muu          $ n          $ radish                 $ N
mun          $ n          $ door                   $ N
munbeob      $ n          $ grammar                $ N
munje        $ n          $ problem                $ N
munhwa       $ n          $ culture                $ N
mul          $ n          $ water                  $ N
mulgeon      $ n          $ thing, item, merchandise$ N
mulron       $ n          $ of course              $ N
migug        $ n          $ U.S.A.                 $ N
migugin      $ n          $ American               $ N
migun        $ n          $ U.S. military personnel$ N
migun        $ n          $ U.S. armed forces      $ N
migeugi      $ n          $ MIG plane              $ N
mis          $ n          $ miss                   $ N
```

misteo $ n $ mister $ N
miyongsil $ n $ beauty parlor $ N
mijangweon $ n $ beauty parlor $ N
mihon $ n $ never married $ N
mihwa $ n $ American money $ N
minyo $ n $ folk song $ N
minganin $ n $ civilian $ N
minsogchon $ n $ Korean Folk Village $ N
mit $ n $ underneath, bottom $ N
mada $ p $ each, every $ N
man $ p $ only $ N
man $ p $ after a lapse of (time)$ N
mankeum $ p $ as.. as.., equal to $ N
myeoch $ adj $ how many $ N
modu $ n $ all $ N
modu $ adv $ all together $ N
modeun $ adj $ all, every $ N
museun $ adj $ what kind of, some kind of$ N
mulron $ adv $ of course $ N
miri $ adv $ in advance $ N
mareu $ v $ be thirsty, become thin$ N
masi $ v $ drink $ N
machi $ v $ finish $ N
maghi $ v $ be blocked $ N
manna $ v $ meet $ N
mandeul $ v $ make, manufacture $ N
manh $ v $ be many/much $ N
mani $ adv $ a lot, plenty $ N
malg $ v $ be clear $ N
maj $ v $ be right, correct $ N
maj $ v $ to fit $ N
maeb $ v $ be hot (in taste) $ N
meog $ v $ eat $ N
meogi $ v $ feed $ N
meonjeo $ adv $ first of all $ N
meol $ v $ be far $ N
meolriha $ v $ keep away from $ N
me $ v $ carry.. on one's shoulder $
myeondoha $ v $ shave $ N
moreu $ v $ not know $ N
mosi $ v $ be with, escort $ N
moeu $ v $ collect $ N
moi $ v $ gather $ N
mojara $ v $ be insufficient $ N
mugeob $ v $ be heavy $ N
mudeob $ v $ be muggy $ N
museob $ v $ be fearful $ N
mud $ v $ ask $
mianha $ v $ be sorry $ N
mid $ v $ believe $ N
mil $ v $ push $ N
madi $ n $ joint $ N
madi $ n $ $
baggat $ n $ outside $ N
bada $ n $ ocean, sea $ N
badasga $ n $ seashore, beach $ N
baji $ n $ pants, trousers $ N
baggyeogpo $ n $ mortar $ N
bagmulgwan $ n $ museum $ N
bagsa $ n $ doctor, Ph.D. $ N
bag $ n $ night(s) $ N
bagg $ n $ outside $ N
ban $ n $ half, class $ N
bandae $ n $ objection $ N
bando $ n $ peninsula $ N
banmal $ n $ rough talk $ N
banaeg $ n $ half price $ N
banchan $ n $ side dish $ N
bal $ n $ foot $ N
balgarag $ n $ toe $ N
baleum $ n $ pronunciation $ N
baljeon $ n $ progress, development $ N
balpyo $ n $ announcement $ N
bam $ n $ night $ N
bab $ n $ cooked rice $ N
basderi $ n $ battery $ N
bang $ n $ room $ N
bangmun $ n $ visit $ N
bangbeob $ n $ method, way $ N
bangeo $ n $ defense $ N
bat $ n $ dry field $ N
bae $ n $ ship, belly, stomach, pear$ N
bae $ n $ times $ N
baegu $ n $ volleyball $ N
baechu $ n $ Chinese cabbage $ N
baechi $ n $ disposition $ N
baetal $ n $ stomach disorder $ N
baeg $ n $ one hundred $ N
baegaggwan $ n $ The White House $ N
baeghwajeom $ n $ department store $ N
beoseu $ n $ bus $ N
beon $ n $ .. time(s), No... $ N
beonyeog $ n $ translation $ N
beonji $ n $ house number suffix $ N
beonho $ n $ number $ N
beojggoch $ n $ cherry flower $ N
byeog $ n $ wall $ N

```
byeong $ n $ disease, bottle $ N
byeonggwa $ n $ branch of the service $ N
byeonggi $ n $ arms, weapons $ N
byeonggigo $ n $ armory $ N
byeongryeog $ n $ military strength $ N
byeongsa $ n $ soldier $ N
byeongweon $ n $ hospital, doctor's office $ N
byeongjang $ n $ E-5, sergeant $ N
bogoseo $ n $ report $ N
bogeub $ n $ supply $ N
bodo $ n $ news report, side walk $ N
bobyeong $ n $ infantry, infantryman $ N
boseogsang $ n $ jewelry store $ N
botong $ n $ regular $ N
bogdeogbang $ n $ real estate agency $ N
bogdo $ n $ hallway $ N
bogsunga $ n $ peach $ N
bogjang $ n $ clothing, attire $ N
bonneteu $ n $ hood, bonnet $ N
bonbu $ n $ headquarters $ N
bolpen $ n $ ball-point pen $ N
bom $ n $ spring $ N
bongtu $ n $ envelope $ N
buawansil $ n $ adjutant's office $ N
bugeun $ n $ vicinity $ N
budae $ n $ troop $ N
bumo $ n $ parents $ N
bubu $ n $ husband and wife $ N
bubun $ n $ part, portion $ N
bueok $ n $ kitchen $ N
buin $ n $ wife, lady $ N
bug $ n $ north, drum $ N
buggoe $ n $ North Korean puppet regime $ N
bughan $ n $ North Korea $ N
bun $ n $ minute, counter of person $ N
bundae $ n $ squad $ N
bunsu $ n $ fountain $ N
gwa $ n $ department $ N
sil $ n $ room, office $ N
bul $ n $ fire, light $ N
bul $ n $ dollar $ N
bulgyeonggi $ n $ depression, recession $ N
bulpyeong $ n $ complaint, grievances $ N
bi $ n $ rain, broom $ N
binu $ n $ soap $ N
bimil $ n $ secret $ N
biseosil $ n $ secretary's office $ N
bihaenggi $ n $ airplane $ N
bihaengjang $ n $ airport $ N
bbalrae $ n $ laundry $ N
bbang $ n $ bread, pastry $ N
bbangjib $ n $ bakery $ N
bu $ n $ section $ N
baro $ adv $ right, exactly $ N
bagge $ p $ nothing but $ N
beolsseo $ adv $ already, so soon $ N
boda $ p $ than $ N
botong $ adv $ usually $ N
buteo $ p $ from $ Y
buneui $ p $ for fractional numbers $ N
bunji $ p $ for fractional number $ N
bbalri $ adv $ fast, quickly $ N
baggu $ v $ exchange $ N
baggwi $ v $ be changed $ N
babbeu $ v $ be busy $ N
bangab $ v $ be delighted to see $ N
bad $ v $ receive $ N
balg $ v $ be bright $ N
balghi $ v $ lighten, reveal $ N
baeu $ v $ learn $ N
beori $ v $ throw away, dump $ N
beonhwaha $ v $ be bustling $ N
beolri $ v $ open, spread $ N
beos $ v $ take(put) off $ N
byeonha $ v $ change $ N
bonae $ v $ send, dispatch $ N
bo $ v $ watch, look at, see $ N
boi $ v $ be visible $ N
boi $ v $ make.. see $ N
budijchi $ v $ bump into $ N
bureu $ v $ call $ N
bureu $ v $ be full $ N
buimha $ v $ be assigned $ N
buiogha $ v $ be insufficient $ N
buchi $ v $ mail $ N
butagha $ v $ ask a favor $ N
bul $ v $ blow $ N
buti $ v $ paste on, attach $ N
bi $ v $ become empty $ N
bibi $ v $ mix, rub $ N
biu $ v $ empty $ N
bil $ v $ borrow $ N
bil $ v $ ask.. pardon $ N
bili $ v $ lend $ N
bbareu $ v $ be fast $ N
bbaji $ v $ be missing $ N
bbalgah $ v $ be red $ N
```

```
bbae $ v $ pull out $ N
byeolro $ adv $ (not) particularly $ N
byeolangan $ adv $ suddenly $ N
beosgi $ v $ peel, strip.. of $ N
sa $ n $ four $ N
sagyeog $ n $ shooting $ N
sagyeogjang $ n $ firing range $ N
sago $ n $ accident $ N
sagwa $ n $ apple $ N
sadan $ n $ division $ N
sarada $ n $ salad $ N
saram $ n $ person, man $ N
saryeonggwan $ n $ commander $ N
saryeongbu $ n $ headquarters $ N
samonim $ n $ madam $ N
samuso $ n $ business office $ N
samusil $ n $ business office $ N
sabyeong $ n $ enlisted man $ N
sasil $ n $ fact, truth $ N
sai $ n $ gap, space $ N
saida $ n $ soda pop $ N
sain $ n $ signature $ N
sajang $ n $ president of a business firm $ N
sajeon $ n $ dictionary $ N
sajeong $ n $ situation $ N
sajin $ n $ photograph $ N
sachon $ n $ cousin $ N
satae $ n $ landslide, state of affairs $ N
sahoe $ n $ society $ N
saheul $ n $ three days $ N
san $ n $ mountain $ N
saneob $ n $ industry $ N
sam $ n $ three $ N
samdeung $ n $ third class/place $ N
samchon $ n $ uncle $ N
sampalseon $ n $ the 38th parallel $ N
sanggong $ n $ air space $ N
sangbyeong $ n $ E-4, corporal $ N
sangsa $ n $ E-8, master sergeant $ N
sango $ n $ A.M. $ N
sangjeom $ n $ store $ N
sangchi $ n $ lettuce $ N
saeg $ n $ color $ N
saengnyeonweolil$ n $ date of birth $ N
saengsan $ n $ production $ N
saengil $ n $ birthday $ N
saenghwal $ n $ life, living $ N
syasseu $ n $ shirt $ N
seo $ n $ west $ N
seorab $ n $ drawer $ N
seoreun $ n $ thirty $ N
seomyeong $ n $ signature $ N
seobanaeo $ n $ spanish (language) $ N
seobiseu $ n $ service $ N
seouldae $ n $ Seoul National Univ. $ N
seouldaegyo $ n $ Grand Seoul Bridge $ N
seoulyeog $ n $ Seoul Railroad Station $ N
seojeom $ n $ bookstore $ N
seogyu $ n $ oil, petroleum $ N
seon $ n $ line $ N
seonmul $ n $ gift $ N
seonsaengnim $ n $ mister, teacher $ N
seonyag $ n $ prior engagement $ N
seonimhasagwan $ n $ first sergeant $ N
seonjeon $ n $ propaganda $ N
seolbi $ n $ facilities, equipment $ N
siseol $ n $ facilities $ N
seoltang $ n $ sugar $ N
seom $ n $ island $ N
seong $ n $ surname $ N
seonggong $ n $ success $ N
seongmyeong $ n $ full name $ N
seongjeog $ n $ grade, score $ N
seongham $ n $ full name $ N
segye $ n $ world $ N
egwan $ n $ custom office $ N
setagso $ n $ laundry $ N
ses $ n $ three $ N
sesjib $ n $ rented house $ N
soegogi $ n $ beef $ N
soqeum $ n $ salt $ N
sonagi $ n $ shower $ N
sodae $ n $ platoon $ N
soryeon $ n $ Soviet Union $ N
soryeong $ n $ major $ N
sori $ n $ sound, voice $ N
sobi $ n $ consumption $ N
sosig $ n $ news $ N
sowi $ n $ 2nd lieutenant $ N
sojang $ n $ major general $ N
soju $ n $ a kind of Korean liquor $ N
sochong $ n $ rifle $ N
sopa $ n $ couch $ N
sopo $ n $ parcel, package $ N
sogdo $ n $ speed $ N
son $ n $ hand $ N
songarag $ n $ finger $ N
```

sonnye $ n $ granddaughter $ N
sonnim $ n $ guest, customer $ N
sonja $ n $ grandson $ N
sonjabi $ n $ handle $ N
songbyeolhoe $ n $ farewell party $ N
sugeon $ n $ towel $ N
sudo $ n $ capital $ N
suribi $ n $ repair cost $ N
surijeom $ n $ repair store $ N
susaneob $ n $ fishery $ N
eoeob $ n $ fishery $ N
sisong $ n $ transportation $ N
suyeom $ n $ beard $ N
suyoil $ n $ Wednesday $ N
siib $ n $ income $ N
supyo $ n $ check $ N
suhag $ n $ mathematics $ N
sugbagbi $ n $ hotel charge $ N
sugsa $ n $ billet $ N
sugje $ n $ homework $ N
sungyeong $ n $ policeman $ N
sunchalcha $ n $ patrol car $ N
sudgarag $ n $ spoon $ N
sul $ n $ liquor/wine $ N
such $ n $ charcoal $ N
sup $ n $ woods $ N
seuweta $ n $ sweater $ N
seupein $ n $ Spain $ N
seungmuweon $ n $ a crew member $ N
si $ n $ o'clock $ N
sigan $ n $ time $ N
siganpyo $ n $ timetable $ N
sigye $ n $ watch $ N
sigol $ n $ countryside $ N
sinae $ n $ downtown $ N
sidae $ n $ era $ N
siabeoji $ n $ woman's father-in-law $ N
sieomeoni $ n $ woman's mother-in-law $ N
sijang $ n $ market, mayor $ N
sicheong $ n $ city hall $ N
siheom $ n $ test, exam. $ N
siggu $ n $ family member $ N
sigdang $ n $ dining hall $ N
sigsa $ n $ meal $ N
sigcho $ n $ vinegar $ N
sinmun $ n $ newspaper $ N
sinbyeong $ n $ recruit $ N
sinbunjeung $ n $ I.D. card $ N
sinho $ n $ signal $ N
sihodeung $ n $ signal lights $ N
sileobyul $ n $ unemployment rate $ N
sib $ n $ ten $ N
ssainpen $ n $ felt pen $ N
ssal $ n $ rice $ N
ssangangyeong $ n $ binoculars $ N
sseuregi $ n $ garbage $ N
sae $ adj $ new $ N
saero $ adv $ newly $ N
seo $ p $ in/at $ Y
seoro $ adv $ each other $ N
se $ adj $ three $ N
sowi $ adv $ so called $ N
swibge $ adv $ easily $ N
sa $ v $ buy $ N
sayongha $ v $ use $ N
sajeongha $ v $ beg consideration $ N
sal $ v $ live $ N
aenggagha $ v $ think $ N
saenggi $ v $ happen, look $ N
seneulha $ v $ be cool/chilly $ N
seo $ v $ stand, stop $ N
seo $ v $ be built $ N
seolmyeongha $ v $ explain $ N
seolsaha $ v $ have diarrhea $ N
seobseobha $ v $ be sorry, sad $ N
seongdaeha $ v $ be grand $ N
se $ v $ count $ N
sesuha $ v $ wash hands/face $ N
seu $ v $ stop, park $ N
suriha $ v $ repair $ N
swi $ v $ rest $ N
swib $ v $ be easy $ N
si $ v $ be sour $ N
siweonha $ v $ be cool/refreshing $ N
sijagha $ v $ begin $ N
siki $ v $ order, cause $ N
siheomha $ v $ experiment $ N
sin $ v $ put shoes/socks on $ N
sid $ v $ load $ N
silh $ v $ be distasteful $ N
silheoha $ v $ disllke $ N
simgagha $ v $ be bland, flat $ N
ssau $ v $ fight, have a quarrel $ N
sso $ v $ shoot $ N
sseu $ v $ write, use, wear (headgear) $ N
sseu $ v $ be bitter $ N
sseui $ v $ be used $ N

```
ssis $ v $ wash $ N
ssisgi $ v $ be washed $ N
ssisgi $ v $ let.. wash $ N
ssib $ v $ chew $ N
ssibhi $ v $ be chewed $ N
agassi $ n $ miss $ N
anae $ n $ wife $ N
adeul $ n $ son $ N
arae $ n $ below $ N
amu $ n $ any $ N
abeoji $ n $ father $ N
abba $ n $ daddy $ N
au $ n $ younger brother $ N
ajeossi $ n $ mister, uncle $ N
ajumeoni $ n $ lady $ N
achim $ n $ morning, breakfast $ N
apateu $ n $ apartment $ N
ahob $ n $ nine $ N
aheure $ n $ nine days $ N
an $ v $ inside $ N
angae $ n $ fog $ N
angyeong $ n $ glasses $ N
anju $ n $ hors d'oeuvres $ N
ap $ n $ the front $ N
ai $ n $ child $ N
ae $ n $ child $ N
agi $ n $ baby $ N
aegi $ n $ baby $ N
yagu $ n $ baseball $ N
yachae $ n $ vegetable $ N
yag $ n $ medicine, drug $ N
yaggug $ n $ pharmacy $ N
yagdo $ n $ sketch map $ N
yagsa $ n $ pharmacist $ N
yagsog $ n $ appointment, promise $ N
yagju $ n $ refined Korean rice wine $ N
yangmal $ n $ socks $ N
yangbog $ n $ Western suit $ N
yangbogjeom $ n $ tailor shop $ N
yangsig $ n $ Western food $ N
yangsigjib $ n $ Western style restaurant $ N
yangjangjeom $ n $ dress shop $ N
yangju $ n $ Western liquor $ N
yanghwajeom $ n $ shoe store $ N
iyagi $ n $ conversation, story $ N
eoreun $ n $ adult, elders $ N
eomeoni $ n $ mother $ N
eojeogge $ n $ yesterda $ N
eoje $ n $ yesterday $ N
eonni $ n $ older sister $ N
eonje $ n $ when, sometime $ N
eolgul $ n $ face $ N
ansaeg $ n $ complexion $ N
eolma $ n $ how much/many $ N
eoleum $ n $ ice $ N
eomma $ n $ mommy $ N
yeoga $ n $ leisure time $ N
yeogaeggi $ n $ passenger plane $ N
yeogwan $ n $ hotel $ N
yeogweon $ n $ passport $ N
yeogi $ n $ here $ N
yeodan $ n $ brigade $ N
yeodeolb $ n $ eight $ N
yeodeure $ n $ eight days $ N
yeoreos $ n $ many people, a large number $ N
yeoreum $ n $ summer $ N
yeosa $ n $ Mrs. $ N
yeobiseo $ n $ female secretary $ N
yeoseos $ n $ six $ N
yeoja $ n $ female $ N
yeohaeng $ n $ travel, trip $ N
yeog $ n $ railroad station $ N
yeongi $ n $ postponement $ N
yeondae $ n $ regiment $ N
yeonragjanggyo $ n $ liaison officer $ N
yeonbyeongjang $ n $ parade ground $ N
yeonse $ n $ age $ N
yeonpil $ n $ pencil $ N
yeonhyu $ n $ long weekend $ N
yeol $ n $ ten $ N
yeol $ n $ fever $ N
yeolsoe $ n $ key $ N
yeolheul $ n $ ten days $ N
yeobseo $ n $ postcard $ N
yeossae $ n $ six days $ N
yeong $ n $ zero $ N
yeonggug $ n $ England $ N
yeongnae $ n $ inside barracks $ N
yeongeo $ n $ English $ N
yeongeobbu $ n $ sales department $ N
yeongha $ n $ below zero $ N
yeonghwa $ n $ movies $ N
yeonghwagwan $ n $ movie theater $ N
yeop $ n $ next $ N
ye $ n $ example $ N
yeoe $ n $ exception $ N
yeeui $ n $ etiquette $ N
```

yejeong $ n $ schedule $ N
o $ n $ five $ N
oneul $ n $ today $ N
obba $ n $ older brother $ N
osjang $ n $ closet $ N
oi $ n $ cucumber $ N
ojeon $ n $ a.m. $ N
ohu $ n $ p.m. $ N
ondolbang $ n $ floor-heated room $ N
os $ n $ clothes $ N
wanhaeng $ n $ slow $ N
wangbog $ n $ round-trip $ N
oegyogwan $ n $ diplomat $ N
oegug $ n $ foreign country $ N
oemubu $ n $ Ministry of Foreign Affairs $ N
oesachon $ n $ maternal cousin $ N
oesamchon $ n $ maternal uncle $ N
oehalmeoni $ n $ maternal grandmother $ N
oehalabeoji $ n $ maternal grandfather $ N
yogeum $ n $ fare, fee $ N
yosai $ n $ nowadays $ N
yosae $ n $ these days $ N
yoil $ n $ days of the week $ N
yog $ n $ abusive language $ N
yogsil $ n $ bathroom $ N
yongmo $ n $ appearance $ N
yongji $ n $ form, stationary $ N
uri $ n $ we $ N
usan $ n $ umbrella $ N
uchegug $ n $ post office $ N
uchetong $ n $ mailbox $ N
upyeon $ n $ mail $ N
upyo $ n $ stamp $ N
undong $ n $ exercise, movement $ N
weon $ n $ won (Korean monetary unit) $ N
weol $ n $ month $ N
weolgeub $ n $ salary $ N
weolmal $ n $ end of a month $ N
weolse $ n $ monthly rent $ N
weolyoil $ n $ Monday $ N
weolcho $ n $ beginning of a month $ N
wi $ n $ top, the upper part $ N
wibanja $ n $ violator $ N
wibyeong $ n $ guard $ N
wicheung $ n $ upstairs $ N
yurichang $ n $ window $ N
yuhaengga $ n $ popular song $ N
yug $ n $ six $ N
yuggyo $ n $ overpass $ N
yuggun $ n $ army $ N
yugsa $ n $ ROK military academy $ N
yugio $ n $ the 6.25 Incident $ N
eumsig $ n $ food $ N
eumsigjeom $ n $ restaurant $ N
eumag $ n $ music $ N
eungjeobsil $ n $ living room $ N
euija $ n $ chair $ N
eumusil $ n $ dispensary $ N
euisa $ n $ medical doctor $ N
i $ n $ tooth $ N
i $ n $ two $ N
i $ n $ person $ N
inam $ n $ South Korea $ N
ideung $ n $ second class/place $ N
ire $ n $ seven days $ N
ireum $ n $ name $ N
imo $ n $ maternal aunt $ N
ibal $ n $ haircut $ N
ibalso $ n $ barbershop $ N
ibeon $ n $ this time $ N
ibyeong $ n $ private (E-2) $ N
ibug $ n $ North Korea $ N
bun $ n $ person $ N
ibul $ n $ Korean quilted blanket $ N
ius $ n $ neighbor(hood) $ N
iyu $ n $ reason $ N
weonin $ n $ cause $ N
itaeri $ n $ Italy $ N
iteul $ n $ two days $ N
ihae $ n $ understanding $ N
ijeon $ n $ before $ N
ihu $ n $ since (then) $ N
ingong $ n $ artificiality $ N
ingu $ n $ population $ N
inmingun $ n $ the People's Army $ N
insagwa $ n $ personnel section $ N
insamcha $ n $ ginseng tea $ N
insang $ n $ impression $ N
il $ n $ work, event $ N
il $ n $ one $ N
il $ n $ -th day $ N
ilgob $ n $ seven $ N
ilgi $ n $ weather $ N
ildeung $ n $ first class $ N
ilbyeong $ n $ private first class $ N
ilbon $ n $ Japan $ N
ilboneo $ n $ Japanese (language) $ N

```
ilbonin $ n $ Japanese (people) $ N
ilsig $ n $ Japanese food $ N
ilyoil $ n $ Sunday $ N
ilyongpum $ n $ daily necessities $ N
ib $ n $ mouth $ N
ibgu $ n $ entrance $ N
yo $ n $ Korean mattress $ N
eunhaeng $ n $ bank $ N
araecheung $ n $ downstairs $ N
yojeueum $ n $ lately $ N
yojeum $ n $ lately $ N
gajeong $ n $ assumption $ N
agga $ adv $ a while ago $ N
ani $ adv $ not $ N
ama $ adv $ probably $ N
aju $ adv $ very $ N
ajig $ adv $ still, yet $ N
yag $ adv $ approximately $ N
eoneu $ adj $ which $ N
eoddeon $ adj $ certain, some $ N
eoddeohge $ adv $ how, somehow $ N
eolmana $ adv $ how much $ N
e $ p $ at, in $ Y
e $ p $ at, in $ Y
e $ p $ at, on $ Y
ege $ p $ to $ Y
egeseo $ p $ from $ Y
eda $ p $ at, in $ Y
eseo $ p $ at, in $ Y
eseo $ p $ from $ Y
egero $ p $ to $ Y
yeogsi $ adv $ also $ N
orae $ adv $ for long $ N
wae $ adv $ why $ N
yo $ adj $ this $ N
i $ adj $ this $ N
i $ p $ $ Y
ina $ p $ or $ N
iramyeon $ p $ when it comes to $ N
irang $ p $ with $ N
ireon $ adi $ such $ N
ireohge $ adv $ like this $ N
iri $ adv $ this way $ N
isang $ p $ above $ N
ije $ adv $ now $ N
inje $ adv $ now $ N
ilbureo $ adv $ on purpose $ N
iljehi $ adv $ altogether $ N
iljjigi $ adv $ early $ N
eui $ P $ $ Y
euro $ p $ with $ Y
euro $ p $ to $ Y
euro $ p $ with $ Y
euroseo $ p $ as $ Y
oe $ n $ besides $ N
eodi $ n $ where, somewhere $ N
an $ adv $ not $ N
areumdab $ v $ be beautiful $ N
apeu $ v $ have a pain $ N
agsuha $ v $ shake hands $ N
annaeha $ v $ guide $ N
anj $ v $ sit down $ N
al $ v $ know $ N
alryeji $ v $ become known $ N
alri $ v $ let.. know $ N
eoryeob $ v $ be difficult $ N
eol $ v $ freeze $ N
eobsae $ v $ get rid of $ N
eobs $ v $ There is no.. $ N
yeonragha $ v $ get in touch with $ N
yeonseubha $ v $ practice $ N
yeonchagha $ v $ arrive late $ N
yeol $ v $ open $ N
eolri $ v $ be opened $ N
eoyagha $ v $ make reservations $ N
o $ v $ come $ N
oreu $ v $ rise, go up $ N
oreu $ v $ climb $ N
ohaeha $ v $ misunderstand $ N
olmgi $ v $ move $ N
unjeonha $ v $ drive $ N
ul $ v $ cry $ N
us $ v $ laugh $ N
weonha $ v $ want $ N
wiheomha $ v $ be dangerous $ N
igi $ v $ win $ N
ireu $ v $ be early $ N
ireu $ v $ arrive $ N
ireu $ v $ tell $ N
yebbeu $ v $ be pretty $ N
isaha $ v $ move $ N
isangha $ v $ be strange $ N
iyongha $ v $ utilize $ N
ihaeha $ v $ understand $ N
ileona $ v $ get up $ N
ileoseo $ v $ stand up $ N
iljeongha $ v $ be regular $ N
```

```
ilchiha          $ v        $ coincide                  $ N
ilg              $ v        $ read                      $ N
ilh              $ v        $ lose                      $ N
lb               $ v        $ put on                    $ N
ibdaeha          $ v        $ join the military service $ N
i                $ v        $ be                        $ N
iss              $ v        $ There is..                $ N
ij               $ v        $ forget                    $ N
usgi             $ v        $ make.. laugh              $ N
insaha           $ v        $ say hello                 $ N
yoguha           $ v        $ demand                    $ N
yocheongha       $ v        $ request                   $ N
ja               $ n        $ ruler                     $ N
agi              $ n        $ self                      $ N
jane             $ n        $ you                       $ N
janyeo           $ n        $ children                  $ N
jadongcha        $ n        $ automobil                 $ N
jari             $ n        $ seat, space               $ N
jasin            $ n        $ oneself                   $ N
jagnyeon         $ n        $ last year                 $ N
jagjeongwa       $ n        $ military operation section $ N
jandon           $ n        $ small change              $ N
ialmos           $ n        $ error                     $ N
jam              $ n        $ sleep                     $ N
jabji            $ n        $ magazine                  $ N
jabchae          $ n        $ name of a dish            $ N
jang             $ n        $ sheet(counter)            $ N
jang             $ n        $ closet                    $ N
janggeori        $ n        $ long distance             $ N
janggwan         $ n        $ cabinet minister          $ N
janggyo          $ n        $ officer                   $ N
janggun          $ n        $ general officer           $ N
jangmo           $ n        $ man's mother-in-law   $  N
jangin           $ n        $ man's father-in-law   $  N
jaeddeoli        $ n        $ ashtray                   $  N
jaemubu          $ n        $ Ministry of Finance   $  N
aemi             $ n        $ interest                  $  N
jeo              $ n        $ I                         $  N
jeogori          $ n        $ jacket                    $  N
jeogi            $ n        $ there                     $  N
jeonyeog         $ n        $ evening                   $  N
jeoheui          $ n        $ we                        $  N
jeog             $ n        $ enemy                     $  N
jeogbyeong       $ n        $ enemy soldier             $  N
jeon             $ n        $ before                    $  N
jeongibul        $ n        $ electric light            $  N
jeongi           $ n        $ electricity               $  N
jeongijul        $ n        $ electric cord             $  N
jeonryag         $ n        $ strategy                  $  N
jeonmunga        $ n        $ specialist                $  N
jeonse           $ n        $ the lease on a deposit base   $  N
jeonse           $ n        $ the proqress of a battle      $  N
jeonjaeng        $ n        $ war                       $ N
jeontu           $ n        $ combat                    $ N
jeontugi         $ n        $ fighter                   $ N
jeonhwa          $ n        $ telephone                 $ N
jeonhawbeonho  $ n          $ telephone number          $ N
jeonhwabeonhobu$ n          $ telophone directory   $ N
jeom             $ n        $ point                     $ N
jeomsim          $ n        $ lunch                     $ N
jeomsimsigan   $ n          $ lunch hour                $ N
jeobsi           $ n        $ dish                      $ N
jeosgarag        $ n        $ chopsticks                $ N
jeonggeojang   $ n          $ railroad station          $ N
jeonggu          $ n        $ tennis                    $ N
jeongryujang   $ n          $ bus stop                  $ N
jeongri          $ n        $ aggangement               $ N
jeongmal         $ n        $ truth                     $ N
jeongmun         $ n        $ main gate                 $ N
jeongbogwa       $ n        $ intelligence section  $ N
jeongbu          $ n        $ government                $ N
jeongsig         $ n        $ a reqular meal, formality $ N
jeongbo          $ n        $ information/intelligence  $ N
jeongjong        $ n        $ rice wine                 $ N
jeongchaek       $ n        $ policy                    $ N
jeongchi         $ n        $ politics                  $ N
je               $ n        $ I                         $ N
jegwajeom        $ n        $ bakery                    $ N
jedog            $ n        $ admiral                   $ N
jeil             $ n        $ the most                  $ N
jejudo           $ n        $ name of a island          $ N
jehan            $ n        $ limitation                $ N
joban            $ n        $ breakfast                 $ N
joseonhotel    $ n          $ Chosun Hotel              $ N
joka             $ n        $ nephew/niece              $ N
jokaddal         $ n        $ niece                     $ N
jongeobweon    $ n          $ employee                  $ N
jongi            $ n        $ paper                     $ N
jonghabcheongsa$ n          $ Combined Government Office $ N
                          $ Bldg
jwaseog          $ n        $ seats                     $ N
jwahoejeon       $ n        $ left turn                 $ N
ju               $ n        $ state, week               $ N
ju               $ n                                $ N
judun            $ n        $ stationing                $ N
jumal            $ n        $ weekend                   $ N
jubyeon          $ n        $ surroundings              $ N
```

```
juso $ n $ address $ N
juyo $ n $ major $ N
juyuso $ n $ gas station $ N
juin $ n $ master of house $ N
juang $ n $ claim $ N
juchajang $ n $ parking lot $ N
junwi $ n $ warrant officer $ N
junjang $ n $ brigadier general $ N
jul $ n $ line $ N
jul $ n $ $ N
junggan $ n $ middle $ N
junggong $ n $ Communist China $ N
junggug $ n $ China $ N
junggugeo $ n $ Chinese language $ N
junggungin $ n $ Chinese people $ N
jungdae $ n $ company (military unit) $ N
jungdaebonbu $ n $ company headquarters $ N
jungdaejang $ n $ company commander $ N
jungryeong $ n $ lieutenant colonel $ N
jungsa $ n $ sergeant first class $ N
jungseobu $ n $ Midwest $ N
jungsun $ n $ middle 10 days of a month $ N
jungsim $ n $ center $ N
jungangcheong $ n $ capital building $ N
jung $ n $ Buddhist Monk $ N
jung $ n $ during $ N
jungwi $ n $ first lieutenant $ N
jungjang $ n $ lieutenant general $ N
junghaggyo $ n $ junior high school $ N
jeunggang $ n $ reinforcement $ N
jeungweon $ n $ increase of the staff $ N
ji $ n $ since $ N
jigab $ n $ purse/wallet $ N
jo $ n $ qroup $ N
jido $ n $ map $ N
jibung $ n $ roof $ N
jiyeog $ n $ area $ N
jipye $ n $ paper money $ N
jihado $ n $ underpass $ N
jihacheol $ n $ subway $ N
jigweon $ n $ staff $ N
jigjang $ n $ place of work $ N
jigweonsil $ n $ staff room $ N
jigwi $ n $ job title $ N
jigchaeg $ n $ duties $ N
jighaeng $ n $ direct going $ N
jilmun $ n $ question $ N
jim $ n $ luggage $ N
jimggun $ n $ porter $ N
jib $ n $ home/house $ N
jibsaram $ n $ wife $ N
jibse $ n $ rent $ N
jjae $ n $ $ N
jjog $ n $ direction, a piece $ N
yujung $ n $ weekdays $ N
jaju $ adv $ frequently $ N
jal $ adv $ well $ N
jeo $ adj $ that $ N
jeori $ adv $ that way $ N
jeoldaero $ adv $ absolutely $ N
jeongmal $ adv $ really $ N
jeongmalro $ adv $ really $ N
je $ adj $ my $ N
jeil $ adv $ most, best $ N
jo $ adj $ that $ N
jogeum $ adv $ a little $ N
jogeumssig $ adv $ little by little $ N
jom $ adv $ a little $ N
juro $ adv $ mainly $ N
jigeum $ adv $ now $ N
jinan $ adj $ past, last $ N
ja $ v $ sleep $ N
jara $ v $ grow up $ N
jareu $ v $ cut S N
jag $ v $ be small $ N
jab $ v $ catch $ N
jae $ v $ measure $ N
jeog $ v $ write down $ N
jeongongha $ v $ major in $ N
joensogdoe $ v $ be transferred $ N
jeolha $ v $ bow $ N
jeolm $ v $ be young $ N
jeomryeongha $ v $ occupy $ N
jeongriha $ v $ put in order $ N
jedaeha $ v $ be discharged from military $ N
jehanha $ v $ limit $ N
josimha $ v $ be careful $ N
joyoungha $ v $ be quiet $ N
jondaeha $ v $ treat with respect $ N
joli $ v $ be sleepy $ N
joleobha $ v $ graduate $ N
job $ v $ be narrow $ N
joh $ v $ be good $ N
johaha $ v $ like $ N
joesongha $ v $ be sorry $ N
ju $ v $ give $ N
judunha $ v $ station $ N
```

```
jumusi $ v $ sleep $ N
jug $ v $ die $ N
jugi $ v $ kill $ N
junbiha $ v $ prepare for $ N
juli $ v $ reduce $ N
jungyoha $ v $ be important $ N
jeulgi $ v $ enjoy $ N
jina $ v $ pass by $ N
jinae $ v $ get along $ N
ji $ v $ lose $ N
jiruha $ v $ be boring $ N
jiki $ v $ guard $ N
jingrubha $ v $ be promoted $ N
jis $ v $ build $ N
jja $ v $ be salty $ N
jja $ v $ organize, squeeze out $ N
jjig $ v $ stamp, take (a picture) $ N
Jalmosha $ v $ make an error $ N
jul $ v $ be reduced $ N
jalri $ v $ be cut $ N
jeog $ v $ be little/few $ N
jol $ v $ doze $ N
bbun $ n $ $ N
cha $ n $ tea, car $ N
chago $ n $ garage $ N
chado $ n $ driveway $ N
chaseon $ n $ lane $ N
chagryug $ n $ landing $ N
chanjang $ n $ cupboard $ N
changgo $ n $ warehouse $ N
changgu $ n $ window $ N
changmun $ n $ window $ N
chaeso $ n $ vegetables $ N
chaeg $ n $ book $ N
chaesang $ n $ desk $ N
cheoeum $ n $ first time $ N
cheon $ n $ thousand $ N
cheolmo $ n $ helmet $ N
cheobboweon $ n $ intelligence man $ N
cheongso $ n $ cleaning $ N
cheoyug $ n $ gymnastics $ N
cheyuggan $ n $ gymnasium $ N
chongmugwa $ n $ administration section $ N
choegeun $ n $ latest $ N
chuseog $ n $ Chuseog Holiday $ N
chuggu $ n $ soccer $ N
chulgu $ n $ exit $ N
chuldong $ n $ mobilization $ N
chulsin $ n $ origin $ N
culjang $ n $ travel duty $ N
cheung $ n $ floor, story $ N
chima $ n $ skirt $ N
chiyag $ n $ toothpaste $ N
chingu $ n $ friend $ N
chincheog $ n $ relative $ N
chil $ n $ seven $ N
chilpan $ n $ blackboard $ N
chimdae $ n $ bed $ N
chimsil $ n $ bedroom $ N
chimib $ n $ invasion $ N
chissol $ n $ toothbrush $ N
cham $ adv $ really $ N
cheoeum $ adv $ first time $ N
cheoncheonhi $ adv $ slowly $ N
cheos $ adj $ first $ N
cha $ v $ be filled up $ N
cha $ v $ be cold $ N
cha $ v $ kick $ N
chari $ v $ prepare $ N
cham $ v $ tolerate $ N
chaj $ v $ look for $ N
chaeu $ v $ fill up $ N
chodaeha $ v $ invite $ N
chughaha $ v $ congratulate $ N
chulgeunha $ v $ go to work $ N
chuldongha $ v $ set out $ N
chulduha $ v $ appear, present $ N
chub $ v $ be cold $ N
chwigeubha $ v $ handle $ N
chwijigha $ v $ get a job $ N
chi $ v $ hit $ N
chiu $ v $ tidy up $ N
chinjeolha $ v $ be kind $ N
chimibha $ v $ invade $ N
chodae $ n $ invitation $ N
chae $ n $ $ N
chinjeol $ n $ kindness $ N
kamera $ n $ camera $ N
kaseteu $ n $ cassette $ N
kaunteo $ n $ cashier's desk $ N
kal $ n $ knife $ N
kaebines $ n $ cabinet $ N
keopi $ n $ coffee $ N
ko $ n $ nose $ N
koteu $ n $ coat $ N
kolra $ n $ cola $ N
keurim $ n $ crean $ N
```

ki $ n $ height $ N
taja $ n $ typing $ N
tajagi $ n $ typewriter $ N
kajasu $ n $ typist $ N
taggu $ n $ ping-pong $ N
tap $ n $ tower $ N
taegug $ n $ Thailand $ N
taepyeongyang $ n $ Pacific Ocean $ N
taegsi $ n $ taxi $ N
taengkeu $ n $ tank $ N
teominal $ n $ terminal $ N
tenis $ n $ tennis $ N
terebi $ n $ television $ N
teipeu $ n $ tape $ N
toyoil $ n $ Saturday $ N
tongsin $ n $ communication $ N
Teureog $ n $ truck $ N
teureongkeu $ n $ trunk $ N
teuggeub $ n $ special express $ N
teugdeung $ n $ special class $ N
teugsil $ n $ special coach/room $ N
teum $ n $ crevice $ N
teum $ n $ spare time $ N
keuge $ adv $ loudly, largely $ N
teugbyeolhi $ adv $ especially $ N
kyeo $ v $ turn on $ N
keu $ v $ be big/large $ N
ta $ v $ get on $ N
ta $ v $ be burned $ N
ta $ v $ put in $ N
taeu $ v $ burn $ N
taeu $ v $ give.. a ride $ N
tonggwaha $ v $ pass through $ N
toegeunha $ v $ go home from office $ N
teugbyeolha $ v $ be special $ N
teugiha $ v $ be peculiar $ N
teul $ v $ turn on $ N
teulri $ v $ be wrong/incorrect $ N
teui $ v $ be cleared $ N
teu $ v $ open $ N
teu $ v $ bud out, dawn $ N
pagoe $ n $ destruction $ N
pachulso $ n $ police box $ N
pati $ n $ party $ N
pal $ n $ eight $ N
pal $ n $ arm $ N
palgun $ n $ Eight Army $ N
pyeon $ n $ flight number $ N
pyeondo $ n $ one-way $ N
pyeonji $ n $ letter $ N
pyeong $ n $ unit of land $ N
pyeongsu $ n $ size of living space $ N
pyeongji $ n $ level ground $ N
po $ n $ cannon $ N
pogyeog $ n $ bombardment $ N
podo $ n $ qrapes $ N
podoju $ n $ grape wine $ N
poro $ n $ POW $ N
pobyeong $ n $ artilleryman $ N
pojang $ n $ pavement, wrapping $ N
poggyeog $ n $ aerial bombing $ N
pogtan $ n $ bomb $ N
pogpung $ n $ stormy winds $ N
pogpungu $ n $ storm $ N
pom $ n $ form $ N
pyo $ n $ ticket $ N
pyojl $ n $ sign $ N
pungseub $ n $ custom $ N
peurangseu $ n $ France $ N
peurangseueo $ n $ French language $ N
buleo $ n $ French language $ N
pihae $ n $ damage $ N
pilripin $ n $ Philippines $ N
pilyo $ n $ need $ N
parah $ v $ be blue $ N
pal $ v $ sell $ N
pyeo $ v $ open $ N
pyeonriha $ v $ convenient $ N
pyeonha $ v $ be comfortable $ N
pobogha $ v $ crawl $ N
pigonha $ v $ be tired $ N
piu $ v $ smoke $ N
palri $ v $ sell $ N
hana $ n $ one $ N
haneul $ n $ sky $ N
haru $ n $ one day $ N
hamul $ n $ luggage $ N
hasa $ n $ staff sergeant $ N
hasagwan $ n $ non-commissioned officer $ N
hao $ n $ p.m. $ N
haggyo $ n $ school $ N
hagnyeon $ n $ grade $ N
hagbi $ n $ school expenses $ N
hagsaeng $ n $ student $ N
hangang $ n $ Han River $ N
hangyeseon $ n $ demarcation line $ N
hangug $ n $ Korea $ N

```
hangugeo $ n $ Korean language $ N
hangugnoegwa $ n $ Korean language dept. $ N
hangugin $ n $ Korean people $ N
hanbog $ n $ Korean costume $ N
hansig $ n $ Korean food $ N
hanjeongsig $ n $ Korean dinner $ N
halmeoni $ n $ grandmother $ N
halabeoi $ n $ grandfather $ N
hamdae $ n $ fleet $ N
haggonggi $ n $ aircraft $ N
hanggu $ n $ harbor, port $ N
hae $ n $ sun, year $ N
haero $ n $ sea route $ N
haegun $ n $ navy $ N
haebang $ n $ liberation $ N
haebyeongdae $ n $ marine corps $ N
haesa $ n $ Naval Academy $ N
haeanseon $ n $ coastline $ N
haeoe $ n $ overseas $ N
handle $ n $ steering wheel $ N
haengjeongbu $ n $ the administration $ N
heonbyeong $ n $ military police(man) $ N
heonbyeongdae $ n $ military police station $ N
hyeongwan $ n $ porch $ N
hyeongeum $ n $ cash $ N
hyeonjae $ n $ present time $ N
hyeong $ n $ style, older brother $ N
hyeongnim $ n $ older brother $ N
hyeongje $ n $ brothers and sisters $ N
ho $ n $ $ N
ho $ n $ $ N
hosu $ n $ lake $ N
hongsu $ n $ flood $ N
hongcha $ n $ black tea $ N
hwarang $ n $ gallery $ N
hwayoil $ n $ Tuesday $ N
hwajangsil $ n $ rest room $ N
hwanyeonghoe $ n $ welcome party $ N
hwanyul $ n $ exchange rate $ N
hwanja $ n $ patient $ N
hwanjeon $ n $ exchange of money $ N
hwaldong $ n $ activity $ N
hwaljuro $ n $ airstrip $ N
hoedam $ n $ conference $ N
hoesa $ n $ company $ N
hoesaeg $ n $ gray $ N
hoeeui $ n $ conference $ N
hoeeuisil $ n $ conference room $ N
hoehwa $ n $ conversation $ N
hunryeon $ n $ training $ N
hwibalyu $ n $ gasoline $ N
hyuga $ n $ vacation $ N
hyugeso $ n $ rest area $ N
hyugesil $ n $ lounge $ N
hyuil $ n $ holiday $ N
hyuji $ n $ tissue paper $ N
hyujeon $ n $ cease-fire $ N
hyujeonseon $ n $ truce-line $ N
hyujitong $ n $ waste basket $ N
heulggil $ n $ dirt road $ N
him $ n $ strength $ N
han $ n $ as long as $ N
hayeoteun $ adv $ anyway $ N
han $ adj $ one $ N
hago $ p $ and, with $ N
honja $ adv $ alone $ N
hamgge $ adv $ together $ N
han $ adv $ approximately $ N
hanpyeon $ adv $ on the other hand $ N
ha $ v $ do $ N
hayah $ v $ be white $ N
heomha $ v $ be rough $ N
hwagsilha $ v $ be certain $ N
hwana $ v $ be angry $ N
himdeul $ v $ be difficult $ N
hwagsinha $ v $ make sure $ N
```