Korean Language Generation in an Interlingua-based Speech Translation System

by

Dennis Woojun Yang

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1995

Author ...
Department of Electrical Engineering and Computer Science
May 26, 1995

Certified ...
Stephanie Seneff
Principal Research Scientist
Thesis Supervisor

Certified ...
Clifford Weinstein
Leader of Lincoln Speech Systems Technology Group
Thesis Supervisor

Accepted by ...
F.R. Morgenthaler
Chairman, Departmental Committee on Graduate Theses

Korean Language Generation in an Interlingua-based Speech Translation System

by

Dennis Woojun Yang

Submitted to the Department of Electrical Engineering and Computer Science
on May 26, 1995, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering

Abstract

Group 24 at MIT Lincoln Laboratory has been developing an automatic speech-to-speech translation system for the English-Korean pair. For the machine translation module, an interlingua system has been adopted. This system analyzes the source language text and represents the results of the analysis in a semantic frame, an unambiguous textual-meaning propositional representation language, from which the text in the target language is generated. For the language generation component, GENESIS, a language generation system developed at the Spoken Language Systems Group at the Laboratory for Computer Science of Massachusetts Institute of Technology, has been utilized. GENESIS has been used for European languages for general purposes and for Japanese in limited domains. It has also been found to be capable of handling some of the linguistic phenomena that are needed for Korean. However, there exist areas in which GENESIS cannot currently handle Korean generation. This thesis explores the degree to which GENESIS is able to manage Korean language generation.

Thesis Supervisor: Stephanie Seneff
Title: Principal Research Scientist

Thesis Supervisor: Clifford Weinstein
Title: Leader of Lincoln Speech Systems Technology Group

Acknowledgments

More than 8 months of research and a bucket of sweat resulted in this document. Without substantial help and support from the people cited below, the completion of this thesis would have not been possible. I would like to take this opportunity to thank the people who have contributed much to the completion of my Master of Engineering thesis.

I wish to express my sincere thanks to my MIT campus supervisor, Dr. Stephanie Seneff. Only with her expertise in natural language processing and her watchful guidance has it been possible for me to start, progress, and finish my thesis. I truely appreciate her technical support and time spent in proofreading my thesis.

I gratefully acknowledge the help of my MIT Lincoln Laboratory supervisor, Dr. Cliff Weinstein. His support ranged from providing all the necessary equipment to proofreading my thesis.

I would also like to recognize MIT Lincoln Lab for allowing me to use its facilities.

A lot of thanks to the Group 24 members of MIT Lincoln Lab for not only helping me do my research, but also for showing me a glimpse of the real world. I would especially like to thank Dinesh Tummala whom I could always count on for guidance on various issues, both technical and personal, and Dr. Youngsuk Lee who shared her Korean expertise with me.

Special thanks to my friends, Shinsuk, Hyunwoo, Dongwhan, and Soojung who scored the performance of the output, and to Jinmo, John and professor Victor Zue who proofread my thesis. A lot of thanks and best wishes for Sunyong who showed me her thesis to look at the format, and for Mira who sent over the material about the Korean grammar.

And, to my parents who immigrated to America to give me the best education they can, I would like to express my deepest gratitude.

Most importantly, I would like to thank God for being with me and keeping me strong in faith throughout the whole process.

The work for this thesis was carried out in the fall of 1994 and in the spring of 1995

at MIT Lincoln Laboratory, Lexington, Massachusetts. This document was written and revised in the spring term of 1995 at MIT Lincoln Laboratory.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Machine translation systems

Group 24 at MIT Lincoln Laboratory has been developing an automatic speech-to-speech translation system (SSTS) for English and Korean. It has been proposed that the English-Korean automatic SSTS be used by military coalition forces in Korea where the need for communication among Korean and American soldiers has been recognized. However, this proposal is difficult to fulfill due to the large differences between the two languages. The purpose of building the English-Korean automatic SSTS is to help the soldiers communicate in their own respective languages.

A typical SSTS works in three phases: speech recognition, language translation, and speech synthesis [1]. The first phase recognizes the speech in the source language (SL) then produces the utterance in text form. The second phase analyzes this utterance and translates it into the target language (TL) in text form. The last phase converts this translation into sound. See figure 1-1.

Most machine translation systems developed to date fall into two categories depending on how the language translation is approached - transfer and interlingua [1]. Transfer systems involve finding the target language correlates for lexical units and syntactic constructions of the source language, whereas in interlingua systems the SL and TL are never in direct contact. Interlingua systems analyze the source language text and represent the results of analysis in interlingua text (ILT), an unambiguous

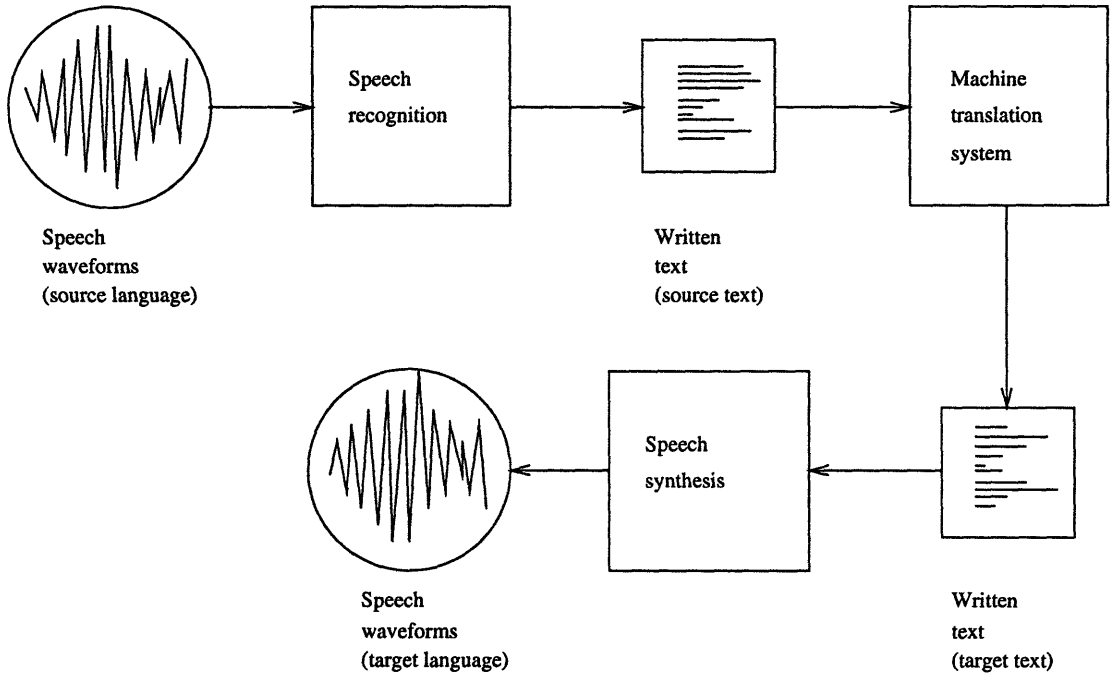

Figure 1-1: A typical SSTS

textual-meaning propositional representation language [2], from which the text in the TL is generated.

The ILT approach has been chosen at MIT Lincoln Laboratory. The system consists of analysis and generation programs [1]. The source language text is processed by a text analysis program. This program uses knowledge of the SL grammar and lexicon to produce ILT. The ILT is passed to the generation program which then produces the output translation in the target language using TL lexicon and grammar. See figure 1-2.

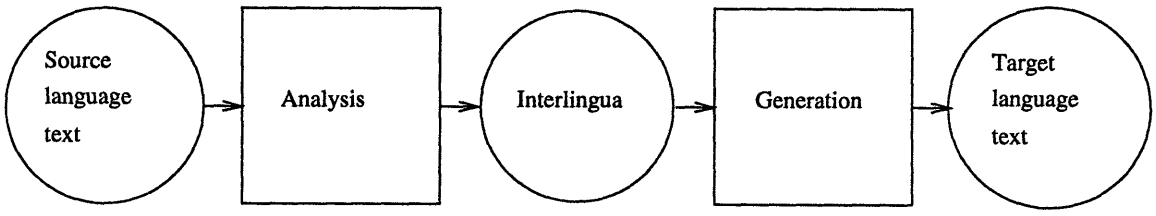

Figure 1-2: Translation using ILT approach [1]

A robust ILT system assumes access to complete knowledge sources for each of the languages the system handles for processing, and it also assumes that IL can adequately represent the semantic meaning of the SL. This assumption is crucial when the approach is applied to some Asian languages such as Korean or Japanese [3]. Those languages have various styles of speech indicating the relative positions, sexes, and ages of the speaker and listener. These differences in the styles can be very complex. Therefore, when those languages are used as TLs, even when a simple English word like "hi" becomes translated into them, the knowledge sources used along with the analysis phase may have to be very complex in order to capture the meaning sufficiently for translation into the TLs. However, for translation within a limited domain, it may be possible to simplify the analysis.

1.2 CCLINC

Common Coalition language at LINColn Laboratory (CCLINC) is a system architecture and concept demonstration for automatic speech-to-speech translation for limited-domain multilingual applications [4]. The proposed application is the coalition battle management environment. The system translates speech in one of three languages (English, French, or Korean) into one of the other two languages or both languages, utilizing a Common Coalition Language (CCL) as a military interlingua [4].

Figure 1-3 depicts the planned structure of CCLINC. The subsystem architecture is composed of a module consisting of speech recognition, natural language understanding, language generation, and speech synthesis for each language. Each of these modules produces a meaning representation in the form of a semantic frame. These semantic frames are transmitted via a Common Coalition Language network to be used as input to the language generator in a different language [4].

The vocabulary, grammar, and semantics of CCLINC are specifically designed to suit brigade communications. A transcription of a Task Force Command Net exercise is being used as the main source in providing a specification of command and control

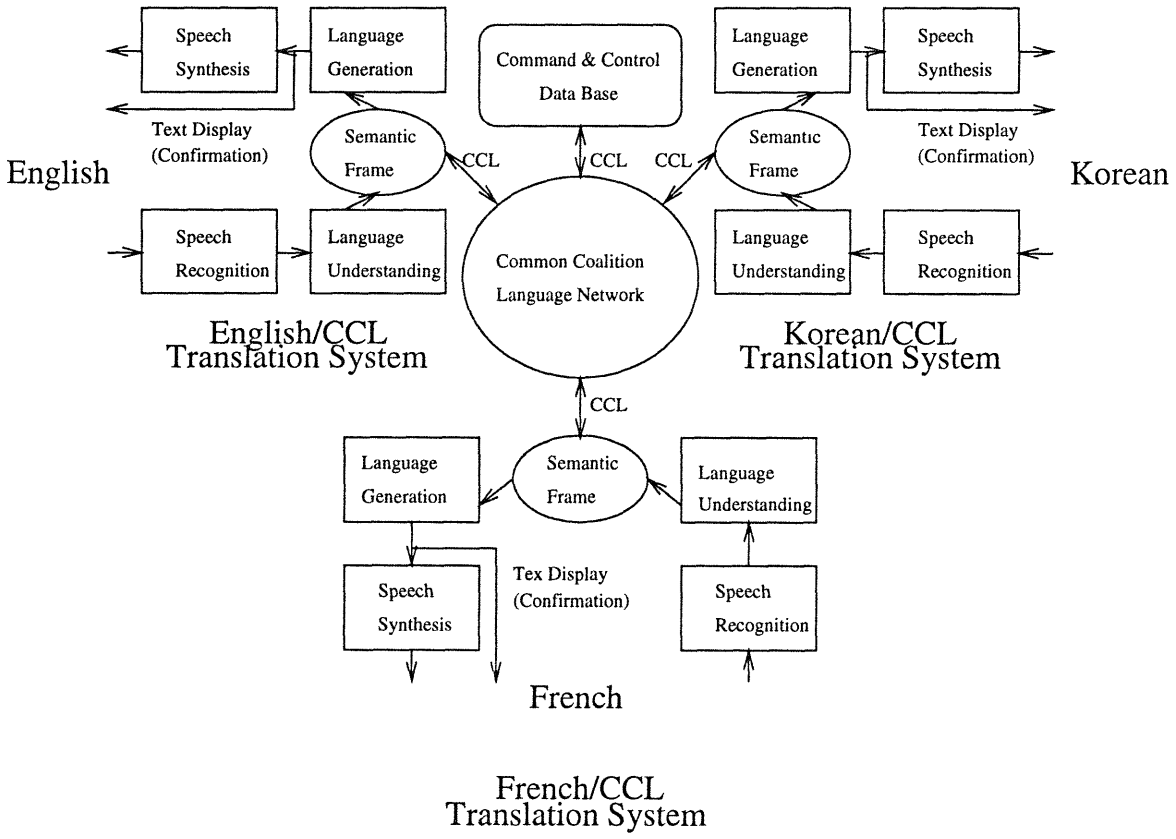

Figure 1-3: System structure for multilingual SSTS [6]

message formats. It contains 1400 utterances, among which 500 sentences have been used both to train and to test the system [4]. Some example sentences from the 500 sentences are given below.

```
(1-a)  Call some artillery
(1-b)  Request permission to defend hilltop echo
(1-c)  Enemy sighted at hilltop charlie
(1-d)  This is delta
(1-e)  Let me get a grid from alpha and I will pass it to you
```

1.3 Korean language in CCLINC

An ideal semantic frame perfectly extracts and represents all the fine details of speech in a source language. Even with this assumption, some difficulties arise in dealing with the Korean language. The most prominent example stems from the fact that there are various styles of speech indicating the relative positions, sexes, and ages of the speaker and listener in Korean. These differences in styles can be very complex. Therefore, when Korean is used as the target language, even a simple English word like "hi" becomes hard to translate. The knowledge sources used along with the analysis phase may have to be very complex in order to capture the meaning sufficiently for translation into Korean, not to mention the need to capture the relative positions, sexes, and ages of the speaker and listener from the context of the source language.

For the proposed task, however, this difficulty may be reduced to a great extent because the domain of usage is very limited. Within this limited domain, it is plausible to assume that the system is to emulate the speech that an educated military male of middle rank would use when talking to his peers.

To illustrate how the ranking of the speaker and listener can affect this simple phrase, consider the following variations of the Korean translation for the same English phrase, "trying to obtain."

```
(3-a) GuHaRyeoNeun JungIYo     - when speaking to a peer.
```

```
(3-b) GuHaRyeoNeun JungIDa    - when speaking to a peer or someone of
                                lower rank.
(3-c) GuHaRyeoNeun JungIYa    - same as above.
(3-d) GuHaRyeoNeun JungIJyo   - when speaking to a slightly older
                                person.
(3-e) GuHaRyeoNeun JungIEoYo  - when speaking to a superior.
(3-f) GuHaRyeoNeun JungIbNiDa - when speaking to a superior.
```

1.4 TINA and GENESIS

For the language parsing component, the Speech Group at the MIT Lincoln Laboratory has decided to use TINA, a language parsing system developed at the Spoken Language Systems Group (SLSG) at the Laboratory for Computer Science (LCS) of the Massachusetts Institute of Technology (MIT) [5]. TINA utilizes key ideas from context free grammars, augmented transition networks, and the unification concept [5]. The details of TINA are beyond the scope of this document.

For the language generation component, a language generation system called GENESIS, also developed at the SLSG [4], has been adopted. GENESIS is driven by three tables: vocabulary, messages, and rewrite rules [4]. They are the parameters for the system which can be manipulated to produce output sentences with a given ILT. By changing these tables, a different set of styles of Korean sentences can be generated from the same English sentence.

GENESIS has been used for European languages for general purposes and for Japanese in limited domains [4]. It has also been found to be capable of handling some of the linguistic phenomena that are needed for Korean. However, there exist areas in which GENESIS cannot currently handle Korean generation. This thesis explores the degree to which GENESIS is able to handle the Korean language generation, and proposes modifications to further generalize GENESIS.

In order to start measuring the expandability of GENESIS to Korean generation, a couple of assumptions have been made. First, it is assumed that the analysis phase of the translation system has been executed correctly and that IL represents the meaning of the SL adequately. Second, Korean generation in a military context gives

the upper bound for the performance of GENESIS as it represents a subset of all the Korean language. The system models the language that educated military personnel of middle rank would use in battlefields.

1.5 Evaluation procedure

A transcription of a Task Force Command Net exercise was used to evaluate the performance of the system. Note that this transcription is the same one that was used to train the system. This decision was made because of the following reason; CCLINC is in its infancy and has numerous deficiencies, and the purpose of this document is not to test how well CCLINC can perform, but rather to identify such deficiencies and give suggestions for improvements. Therefore, it was necessary to examine how CCLINC behaved with the training data.

The parsed sentences are evaluated based on how closely the meaning has been preserved (adequacy) and how fluent the translation sounds (fluency). This evaluation was carried out by four native Korean speakers, who scored each translation from 5 to 1, 5 being the best and 1 being the worst.

Chapter 2

Korean language phenomena

2.1 Word order rules

The basic word order in Korean is characterized by Subject-Object-Verb (SOV), clearly different from Subject-Verb-Object (SVO) of English. Some examples of distinguishable characteristics of Korean word order are [3]:

1. The verb comes at the end of a clause.
2. Negation is represented by changes at the ending of the verb.
3. Noun phrases are followed by postpositions, unlike English where noun phrases are preceded by prepositions.
4. Modifiers precede the words they describe.
5. Words that need to be emphasized are usually put close to the verb.
6. When word A modifies word B and word C modifies word D, the two pairs must not cross each other. The word order A C B D violates this rule, since A-B crosses C-D. However, A C D B satisfies this rule.

Rule 1, along with rule 3, are the basic characteristics of Korean that work with the properties of postpositions to allow a wide variety of sentences having essentially the same lexical meaning, but provoking subtly different contextual meanings. Having the verb come at the end allows space in which all the preceding words can be scrambled with each other in front. This scrambling, however, does not give rise to

any confusion, as postpositions clearly identify which word is fulfilling which role in a particular sentence. In light of this fact, it will be necessary to explain what the Korean postpositions do.

The postpositions serve similar functions that English prepositions have: they describe the relationships immediately preceding nouns/noun phrases have with other words in the clause/sentence. The description of this function can easily be found in any literature discussing Korean grammar. The following description is a translation of essential points made by Cho [8].

Cho defines postpositions as "words that do not have independent meanings of their own, but, when attached to other words, give them grammatical relationships with the rest of the words or additional meanings." Some of the prominent properties of postpositions are as follows [8]:

1 Since only postpositions do not have independent meanings of their own in Korean, they can be distinguished from all other classes of words.

2 They are usually put at the end of nouns, adverbs or other postpositions.

(2-a) JaJeonGeo''Reul'' SassDa - following a noun
BICYCLE BOUGHT
(I) bought a bicycle

(2-b) NalSsiGa MobSi''Do'' NaBbeuJi? - following an adverb
WEATHER VERY BAD ?
The weather is very bad, isn't it?

(2-c) DangSin''GgaJi''''Ga'' HabGyeogIRaNe - following a postposition
YOU UP TO ACCEPTED
Those who are accepted are up to and including you

When the attachment happens, the preceding words do not alter their endings, unless they are pronouns. Even pronouns do not always change their endings.

(2-d) Na ''Ga'' --> NaiGa - vowel ''a'' changed to ''ai''

```
      I          I

(2-e) Na ''Neun'' --> NaNeun - no change
      I            I
```

3 Some sets of postpositions have identical meanings, but are used differently depending on the ending of the preceding syllable, i.e., whether the ending is a vowel or a consonant. The following examples illustrate this property. "Reul" and "Eul" have identical meaning, that is, they indicate that the preceding word is a direct object. However, "Reul" is used when the ending of the preceding word is a vowel, whereas "Eul" is used when the ending is a consonant.

```
(2-f) Neo''Reul'' - vowel ending ''eo''
      YOU

(2-g) Chaig''Eul'' - consonant ending ''g''
      BOOK
```

4 Some postpositions, such as the ones that mean "of" and "be", can be omitted without altering the meaning of the phrase, resulting in some compactness. This omission may also occur when omitting does not confuse any grammatical relationships among the words in a sentence.

```
(2-h) URi''Eui'' NaRa --> URiNaRa
      OUR      NATION   OUR NATION

(2-I) IGeosEun YeoJa''I''''Go'' JeoGeosEun NamJa''I''''Da''
      --> IGeosEun YeoJa  ''I''''Go'' JeoGeosEun NamJa''Da''
          THIS     FEMALE BE  AND THAT       MALE  BE
      This is a female, and that is a male

(2-J) NeoNeun SugJe''Reul'' Hai --> NeoNeun SugJe    Hai
      YOU     HOMEWORK    DO       YOU     HOMEWORK DO
      You do the homework
```

5 Postpositions can be classified into three categories: conjunctive, complementary, and role-assigning. Conjunctive postpositions are similar to the English word "and"

in a sense that they connect the preceding and the following words, which share a common property, into a group. Complementary postpositions add special meanings to the preceding words, such as comparing, lower/upper bounding, all-including, beginning, ending, selecting, limiting. Role-assigning postpositions assign roles (subject, object, etc) to nouns/noun phrases. This group of postpositions will be explained further as English has no such equivalents.

Role-assigning postpositions have the following properties.

1. Role-assigning postpositions follow nouns, noun phrases, and gerunds.
2. Roles that can be assigned and the postpositions that assign those roles are as follows:

```
subject          - I/Ga, Nuen/Eun, GgeSeo, ESeo, Seo
direct object    - Eul/Reul
indirect object  - I/Ga
possessive       - Ui
adverb           - E, EGe, HanTe, URo
calling          - A/Ya, IYeo
verb             - IDa ''be''
```

Rule 2 will be explored further when discussing how Korean verbs behave.

Rule 4 reveals the most distinct characteristics of ordering in Korean. In English for example, modifiers can follow the clauses that they modify; Korean modifiers always precede the clauses they modify. Consider this example.

(2-k) WAITRESS SERVING POTATO CHIPS

would be translated as:

(2-l) POTATO CHIPS SERVING WAITRESS

Rule 5 needs special attention, as it allows a wide variety of word ordering. Consider the following examples.

(2-m) Na Neun ONeul 2Si E HagSaingHoiGoan ESeo ChinGu Oa
I TODAY 2 O'CLOCK AT STUDENT CENTER AT FRIEND WITH

JeomSim Eul MeogEossDa.
LUNCH ATE.

(2-n) Nai Ga 2Si E HagSaingHoiGoan ESeo ChinGu Oa JeomSim
I 2 O'CLOCK AT STUDENT CENTER AT FRIEND WITH LUNCH

Eul MeogEunGeos Eun ONeul IEossDa.
EATING TODAY WAS.

(2-o) Nai Ga ONeul HagSaingHoiGoan ESeo ChinGu Oa JeomSim Eul
I TODAY STUDENT CENTER AT FRIEND WITH LUNCH

MeogEunGeos Eun 2Si EossDa.
EATING 2 O'CLOCK WAS.

(2-p) Nai Ga ONeul 2Si E ChinGu Oa JeomSim Eul MeogEunGeos
I TODAY 2 O'CLOCK AT FRIEND WITH LUNCH EATING

Eun HagSaingHoiGoan ESeo YeossDa.
STUDENT CENTER AT WAS.

(2-q) Nai Ga ONeul 2Si E HagSaingHoiGoan ESeo JeomSim Eul
I TODAY 2 O'CLOCK AT STUDENT CENTER AT LUNCH

MeogEunGeos Eun ChinGu Oa YeossDa.
EATING FRIEND WITH WAS.

(2-r) Nai Ga ONeul 2Si E HagSaingHoiGoan ESeo ChinGu Oa
I TODAY 2 O'CLOCK AT STUDENT CENTER AT FRIEND WITH

MeogEunGeos Eun JeomSim IEossDa.
EATING LUNCH WAS.

(2-m) can be translated to "I had lunch with a friend at the student center at 2 o'clock today." The subsequent sentences place special emphasis on the words "today", "2 o'clock", "student center", "friend", and "lunch", respectively, by putting them close to the verb. It bears mentioning that postpositions in Korean are what

make this scrambling possible, while still preserving the essential meaning of the original sentence and the role each of the word satisfies.

2.2 Conjunctive relations

Conjunctive relation can be divided into two parts; temporal and logical [8]. The temporal relation determines how events should be ordered by using words like "after," "before," "during," "lead to," "result," and "then." The logical relation determines the logical connections among the events.

2.3 Verb suffixes

Perhaps the verbs of the Korean language are what distinguish Korean from all other languages. A great number of variations of the suffixes with slight and subtle differences in meaning among them mark not only past, present, and future tenses as in English, but also indicate other traits like politeness, and degree of familiarity of the speaker with respect to the listener [8]. The honorific/polite suffixes are discussed first in this section.

2.3.1 Honorific/polite suffixes

1 When the subject of a sentence is of a higher rank than the speaker, in order to show respect to the subject, honorific suffixes are added to the verb. "Si" is one of the most widely used honorific suffixes.

```
(2-s)  EoMeoNiGgeSeo JinJiReul Deu''Si''EossDa
       MOTHER        MEAL      ATE
       (My) mother ate the meal
```

When "Si" is combined with another postposition "Ob," it becomes an even stronger honorific suffix.

```
(2-t)  ImGeumNimGgeSeoNeun SuRaReul Deu''Si''''Ob''SoSeo
       KING                MEAL     EAT
       Please eat the meal, (my) king
```

Note that "JinJi" and "SuRa" mean the same, but are used differently. "JinJi" is already an honorific noun for "Bab" (meal) in Korean, but "SuRa" is so honorific that it is only used when referring to the meals of a king. This honorific style matches with the use "SiOb" in the example (2-t).

Consider the following example.

```
(2-u)  DongSaingI      NajJamEul JanDa
       YOUNGER BROTHER NAP       SLEEP
       (My) younger brother is taking a nap

(2-v)  SeonSaingNimGgeSeoNeun NajJamEul JuMu''Si''nDa
       TEACHER                NAP       SLEEP
       (My) teacher is taking a nap
```

"JanDa" means "to sleep." And adding "Si" to it alters it into "JuMuSinDa," the honorific form of "JanDa."

2 When an honorific/polite verb also indicates tense, honorific-tense-polite is the order that the respective endings follow. For example, consider the word "eat."

```
Lexical          - MeogDa
Honorific        - JabSu''Si''Da
Past Honorific   - JabSu''Si''''Eoss''Da
Future Honorific - JabSu''Si''''Gess''Da
Polite           - JabSu''Si''''GessSaO''IDa
```

The citation form of "eat" is "MeogDa." Adding "Si" transforms it to "JabSuSiDa." Adding "Eoss" on top of the honorific form makes it past honorific, whereas adding "Gess," makes it future honorific. Furthermore, the honorific form with "GessSaO" becomes the polite form.

2.3.2 Tense suffixes

1 The three basic tenses are past, present, and future tenses. To indicate these tenses, "Eoss/Ass," "n/Neun" and "Gess" are added, respectively. Again, let us consider the word "eat."

```
Past    - Meog''Eoss''Da
Present - Meog''Neun''Da
Future  - Meog''Gess''Da
```

2 These tenses may be superimposed as in the following examples.

```
(2-w)  JiGeumJjeumEun MulGoGiReul Jab''Ass''''Gess''Da
       BY NOW         FISH        HAVE CAUGHT
       (He) must have caught a fish by now

(2-x)  GeuDdaiNeun MulGoGiReul Jab''Ass''''Eoss''Da
       THAT TIME   FISH        CAUGHT
       (I) caught a fish at that time
```

3 "Gess" is used to mean both "shall" and "will."

2.3.3 Type-defining suffixes

1 Some of the widely used types of sentences in Korean include statements, exclamations, interrogatives, commands, and requesting sentences. These are completely analogous to their English counterparts. Again, let us use the word "eat" to demonstrate them.

```
(2-y)  AGiGa BabEul Meog''NeunDa''
       BABY  MEAL   EATING
       The baby is eating a meal

(2-z)  AGiGa BabEul Meog''NeunGuNa''
       BABY  MEAL   EATING!
       The baby is eating a meal!
```

(2-aa) AGiGa BabEul Meog''NeuNya''
BABY MEAL EAT?
Is the baby eating a meal?

(2-ab) AGaYa BabEul Meog''EoRa''
BABY! MEAL EAT
Eat the meal, baby.

(2-ac) AGaYa BabEul Meog''Ja''
BABY! MEAL LET'S EAT
Let's eat the meal, baby

2 Roughly speaking, there are two kinds of verbs. One is called DongSa, and these verbs are ones that describe movements of human, animals, etc. The other one is called HyeongYongSa, which describe states of objects. In English, the latter is not classified as verbs, but as adjectives with the verb "be." Words such as "be beautiful," "be large," "be hungry" are two-word verbs composed of the "be" verb and an adjective in English, but in Korean. they are simply one-word verbs.

With HyeongYongSa, some limitations are imposed regarding what types of sentences are possible. HyeongYongSa can not be used for commands and requesting sentences. Furthermore, "ARa/EoRa" are used to make exclamations when they are attached to HyeongYongSa, whereas they make commands when attached to DongSa.

2.3.4 Conjunctive suffixes

1 Conjunctive suffixes that enumerate complementing phrases are "Go," "Myeo," "MyeonSeo."

(2-ad) JeonHoaHa''MyeonSeo'' TVReul BonDa
TELEPHONE TV WATCH
(I) am watching TV while talking on the phone

2 Those that enumerate opposite phrases are "GeoNa"/"GiNa," "DeunJi"/"DeonJi," "GeoNi," "NeuNi."

(2-ae) Ga‘‘DeunJi’’ Mal‘‘DeunJi’’ Ne MaEumDaiRo HaiRa
GO NOT YOUR MIND PLEASES DO
You decide whether to go or not to go as you desire

3 There are conjunctive suffixes that relate the preceding and the following phrases in specific ways.

"Na," "JiMan" are used to mean the English equivalent "although."

(2-af) ManhI Jass‘‘JiMan’’ AJigDo JolRiDa
ALOT SLEPT STILL SLEEPY
Although (I) have slept alot, I am still sleepy

"RyeoGo," "Ryeo" are equivalent to "in order to."

(2-ag) IlJjig Ggae‘‘RyeoGo’’ IlJjig JassDa
EARLY GET UP EARLY SLEPT
(I) went to bed early to wake up early

"NeuRaGo," "ASeo"/"EoSeo," "AYa"/"EoYa," "GeoMan"/"GeoNiOa" are used to indicate that the preceding phrase is the cause of the following phrase.

(2-ah) BiGaW‘‘aSeo’’ USanI PilYoHaissDa
RAINING UMBRELLA NEEDED
(I) needed an umbrella because it was raining

"nDe" is used when describing the background that will be used for the following phrase.

(2-ai) SimSimHa‘‘nDe’’ MuEossEul HalGga?
BORED WHAT DO
(I) am bored. What shall (I) do?

2.3.5 Gerund suffixes

1 Suffixes such as "m" and "Gi" make gerunds out of verbs.

(2-aj) GeuReul DdaRaHa‘‘m’’Eun HimDeulDa
HIM FOLLOWING HARD
It is hard to follow doing what he does

(2-ak) GuaJaReul Meog‘‘Gi’’Ga SilhDa.
COOKIES EATING DISLIKE
(I) dislike eating the cookies

In (2-aj) the gerund serves as the subject of the sentence, and the gerund in (2-ak) serves as the direct object of the sentence.

2 Very frequently, “n Geos” is used to form a gerund. This form has an identical meaning as the cases “m” and “Gi,” but provides more flexibility in using the gerund. This form is used more often in colloquial language.

(2-al) GuaJaReul Meog‘‘Neun’’ ‘‘Geos’’I SilhDa.
COOKIES EATING DISLIKE
(I) dislike the eating the cookies

Some of the distinctive Korean language phenomena have been explored in this chapter. The next chapter discusses how these phenomena could be implemented in a language generator called GENESIS.

Chapter 3

GENESIS

GENESIS is a language generator which produces well-formed sentences from a semantic representation. GENESIS paraphrases the semantic representation of English sentences, and, as used here, generates Korean sentences from it. Before discussing the mechanism of GENESIS, it is necessary to look at the structure of its input; a semantic frame.

3.1 Semantic frames

The meaning representation that is used as input to GENESIS is called a semantic frame. The semantic frame ideally captures the meaning of the speech in the source language with the hierarchical dependencies among the parts of the speech preserved. The semantic frame recognizes that sentences are composed of clauses, topics and predicates [6]. Note that "predicate" includes adjectives and prepositional phrases, as well as verbal predicates. See the semantic frame of a sample sentence below. The corresponding parse tree is shown in figure 3-1.

```
Input:  Request permission to defend hilltop echo

Semantic Frame (CCL)
{c statement
   :mode ''fpl''
```

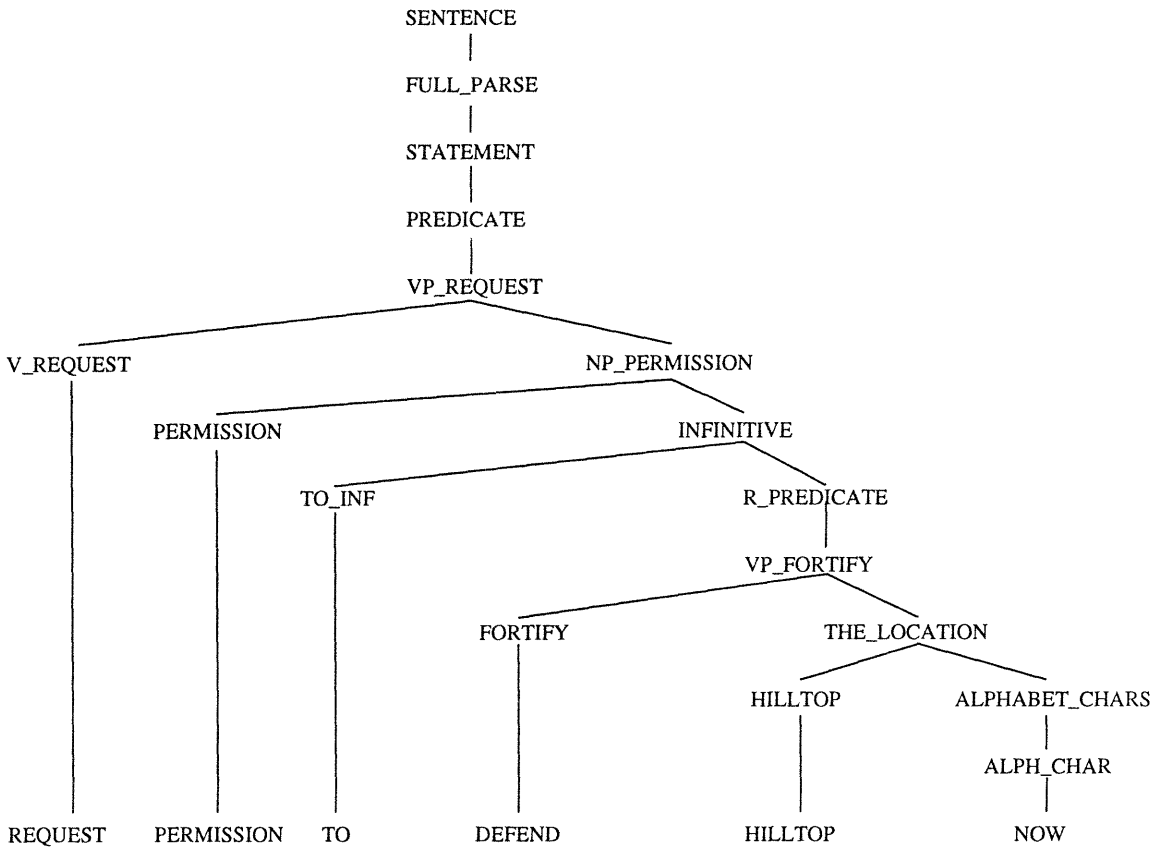

Figure 3-1: Parse tree for a sample sentence

```
:number ''fpl''
:pred {p v_request
        :topic {q permission
                  :complement {p fortify
                                 :aux ''to''
                        :topic {q hilltop
                           :pred {p initials
                                 :topic ''echo''}}}}}}
```

The structure and the function of the semantic frame will be discussed in more detail when discussing how GENESIS uses the semantic frame.

3.2 Mechanism of GENESIS

There are two major parts to GENESIS. One part is the kernel of GENESIS which does not change with respect to the target language, and the other part is the shell of GENESIS which realizes the output sentences, and is therefore target-language-dependent [6]. The kernel is the engine of the system that paraphrases the semantic frame, and generates the output by utilizing the information about the target language embedded in the latter part. The shell specifies the characteristics of the target language with three modules: a lexicon, a set of messages, and a set of rewrite rules [6]. The mechanism of the engine will be implicitly described when discussing the details of each of the modules. Note that since the semantic frame is encoded in English, entries in the lexicon and the set of messages are expressed in English. This, by no means, implies that English is the most proper language for semantic representation, but it is chosen only for the sake of convenience, as most engineers using the system can understand English.

3.2.1 Lexicon

The lexicon associates each semantic frame entry with its corresponding form in the target language [6]. This mapping takes various linguistic phenomena such as inflections into consideration [6]. Table 3.1 shows an example lexicon for English.

Table 3.1: Example lexicon entries for English

V	V	"Verb" THIRD "es" ROOT "e" ING "ing"...
N	N	"NOUN" PL "s"
be	X	"be" ROOT "be" THIRD "is" ING "being"...
do	X	"do" THIRD "does"...MODE "root"...
will	X	...MODE "future"...
2	D	"two" CARDINAL "second"

Each entry in the lexicon has a name described by the part of speech tag (e.g., N (Noun), PREP (Preposition), V (Verb)), a stem, and various derived forms. Part of speech entries specify the default endings to the entries whose morphological variants are regular. For example, a typical noun (N) in English becomes plural when an "s" is attached to the end. These default values can be overruled by explicit lexical entries, as in the English verbs "be" and "do."

Each entry can have its own grammatical specifications that are needed to produce a correct lexical form [6]. To illustrate this, consider the fact that the Korean language has two different ways of reading Arabic numbers. One uses Korean, and the other uses Chinese. When the latter is used, the pronunciation is not identical to the pronunciation that Chinese people use today. The Chinese reading is used mostly in ordinary usage such as mathematical terms, telephone numbers, or room numbers. However, for counting something with an order (cardinal numbers), or for people's ages, Korean is used. Another example is that auxiliary verbs set the mode of the main verb, as "will" in English will set the mode of the main verb to be "root." Whether a particular entry is to be treated as a verb or as an adjective is controlled in the lexicon and can be language dependent. This particular feature is especially relevant for Korean, as many adjectives become verb-like in main clauses.

3.2.2 Messages

Messages are grammar templates of the target language that control the ordering of the parts of speech [6]. The topics, predicates, and clauses of a semantic frame get

transformed into phrases of the target language recursively according to this set of grammar templates [6]. A typical template consists of a message name and a sequence of words and/or keywords that describe the message. Words can be inserted before or after any of the keywords, and a default value can be specified when a keyword has no value.

3.2.3 Rewrite rules

The rewrite rules are intended to handle the linguistic phenomena that are hard to deal with through the mechanisms of lexicon and messages [6]. The typical phenomena are phonotactic constraints and contractions. For example, rewrite rules can be used to choose the correct form of the indefinite article "a" or "an," or to merge "a other" into "another." The flexibility of the rewrite rules is not limited to these and will be explored further when discussing how they are used in the case of Korean.

3.3 GENESIS for Korean

Appendices A, B, and C contain the files for lexicon, messages, and rewrite-rules, respectively.

3.3.1 Lexicon

The lexicon has nine distinct linguistic subcategories: adjectives, conjunctions, auxiliary verbs, clauses, determiners, nouns, pronouns, adverbs, and verbs. For entries in each of these subcategories, a list is provided which enumerates the category names of the semantic frame along with their counterparts in the target language. In other words, this lexicon functions as if it were a bilingual lexicon that is used in a typical transfer translation system between two languages, except for the fact that the source lexicon being used is derived from the semantic frame rather than from a raw text string.

Many English words are lexically ambiguous in the sense that they have multi-

ple meanings. For example, the adjective "heavy" could mean having great weight, hard to bear, serious, profound, difficult, and so on. In English, although the single word "heavy" can capture all these different meanings, each of these meanings must be mapped to different semantic frame words so that they can be distinguished appropriately in the target language. Unfortunately, due to an insufficient number of training sentences, only a small portion of such meanings having the same English words have been incorporated into the lexicon. In cases where a single semantic frame word has more than one meaning, the one that is most likely to be used in the military context has been chosen to be the Korean equivalent, i.e., "DaeGyuMoEui" meaning "large-scaled" has been chosen.

It is possible for two different semantic frame adjectives to have one Korean equivalent. Unlike the case above, this does not create much trouble as the precise meaning reveals itself from the context of the translated Korean sentences.

The Korean language does not have articles such as "the" or "a." Nonetheless, occasions arise when one needs to include the meaning of the articles explicitly. "A" can be translated to "HaNaEui" or the contraction form "Han" in such cases. A typical Korean speaker would use the contraction form in his speech. When "HaNaEui" is used, it directly describes the following noun with the meaning of "one," as in "HaNaEui Chaeg" meaning "one book." "Han" functions a bit differently from "HaNaEui." When "Han" is used, a counting noun always follows. For example, "one book" can be translated to "Chaeg HanGueon." Here, "Gueon" is the counting noun, designated specifically for counting the number of books.

One unsolved problem with adjectives stems from the fact that they can be used to describe nouns and can also be used in variation with the "be" verb to describe a state. This may not cause any problems if the target language uses adjectives in the same way, but Korean is not such a language. For one thing, Korean does not have linking or auxiliary verbs. Before suggesting a possible solution to the above problem, it is necessary to describe how "be" verbs can be reflected in Korean verbs, and the kinds of verbs that Korean has.

"Be" verbs have at least four different functions in English. The first one is used

to indicate the existence of an object, as in "there is a book on the table." The second one is used to indicate two things equal in meaning. "God is love" is such an example. The third one is used with past participles of intransitive verbs as an auxiliary verb. Finally the last one is used with adjectives to describe the state of an object. Korean handles each of these four cases differently. This handling is done by manipulating the endings of related verbs. A system which translates from an English system into the appropriate "be" auxiliary for Korean would have to tie the "be" verb in the English sentence to the correct "be" inflections for the related verbs.

There are two kinds of verbs in Korean: action verbs, and adjectival verbs. Action verbs behave just like their counterparts in English. They simply express acts and occurrences. However, adjectival verbs are verbs that describe the mode of being, and are equivalent to adjectives with "be" verbs in English. In other words, Korean has special verb endings to handle the first three functions of "be" verbs. These are "IssDa," "IDa," and "EossDa." These verb endings cover the three roles along with some changes within the roots of verbs. However, Korean does not have a simple verb describing the state of an object by using adjectives. Instead, it has adjectival verbs. In other words, a phrase like "is pretty" is considered as one verb in Korean, and can be translated to "GobDa." These adjectival verbs do not have as many complex verbal endings as action verbs have, as discussed in Chapter 2.

Conjunctions do not pose as much difficulty as adjectives. However, "and" can be lexically ambiguous. It can be used when enumerating things, or when connecting two parallel clauses. Korean has two different words for these, "Oa," "Goa," or "HaGo" for the former, and "GeuRiGo" for the latter. Again, a distinction between the two cases is needed. Note that the ambiguity would not blur the meaning of the translation. It will only decrease the fluency of the translation.

Korean does not have specific linking verbs or auxiliary verbs. Each verb has its own variety of endings which carry the meaning that linking or auxiliary verbs are designed to deliver. As a consequence, these verbs of the semantic frame do not get mapped into any Korean words. They only specify the mode of the verb in order to specify the proper ending of the verb. For example, the auxiliary verb "will" sets the

mode of the corresponding Korean verb to be "future." This "future" mode then is used when selecting the proper mode of the verb later on. This selection process will be discussed later in this section.

Clause-level semantic frames can also set the mode. (See the entries for "command1" and "command2" in Appendix A) These do not share any similarities with auxiliary verbs, but nonetheless, are useful signals when selecting proper Korean verb endings. Note that the command has two different modes. "Command1" refers to imperative sentences as in "to direct authoritatively." "Command2" is used for suggesting sentences as in "let's do..."

Most determiners can be directly mapped with equivalent Korean words without much lexical ambiguity. Numbers are included in this category. In Korean, as well as English, there is a distinction between counting numbers and cardinal numbers. The biggest difference is that Chinese pronunciation is used for counting numbers and pure Korean pronunciation is used for cardinal numbers. For most items that need numbering, including mathematics, Chinese pronunciation is used. The exceptions are cardinal numbers and ages of people, in which pure Korean pronunciation is used. A syllable "Jjae" is attached to form cardinal numbers; this "Jjae" is similar in role to " th" in English.

Nouns also have lexical ambiguity, just as adjectives do. Some nouns, such as "eagle" or "east" have straightforward equivalents in Korean, but most nouns do not. When choosing the mapping words among many possible choices, the ones that would most likely be used in a military context are chosen. One such example would be the word for "terrain." There are at least three Korean translations for this word: "Ji Hyeong," "JiSe," and "JiYeog." Among these translations "JiYeog" has been chosen as it seemed to be the choice that would most likely be used in a military context. When a semantic frame word has multiple translations and is not a military term, the translation that would most likely be used by educated civilians in daily life has been chosen.

Pronouns can be straightforwardly mapped. Each semantic frame pronoun has a Korean equivalent and one piece of additional information which indicates what is

called "NUM." "NUM" tells whether the pronoun is first person, second person, or third person.

The current lexicon contains only three adverbs, which have a simple mapping.

Sohn categorizes Korean verbs into eight distinct groups distributed in three broad classes: four kinds of regular-consonant-final group, three kinds of irregular-consonant-final group, and one kind of vowel-final group [9]. This classification was the one that the initial system's verb classification was based on, but it later proved inadequate. Although Sohn's approach might be linguistically exhaustive, it omitted quite a few classes of verbs and oversimplified the classification to be used for this project. The current system setup uses a modified version of Sohn's classification.

The classification is based on how verbal endings change when the verbs are used in different kinds of sentences: present tense sentences (first person singular, second person singular, third person singular, first plural, second plural), present continuing tense sentences, future tense sentences, command sentences, requesting ("let's do...") sentences, case clauses, and infinitive phrases.

These cases are certainly not exhaustive and are even redundant for Korean. For example, interrogative sentences or exclamation sentences are not being considered. Also, Korean does not distinguish among first, second, or third person. Furthermore, singular and plural sentences use the same verbal endings. As these facts show, Korean verbs do not capture all the features of English verbs. This, however, does not mean that Korean language generation is simple. Korean verbs have many linguistic phenomena that English verbs do not have, and this presents the most difficult problem in Korean language generation from interlingua. Before discussing this problem, the features of verb classification and its structure are discussed below.

The most noticeable difference between the modified classification and the original one is the addition of "HaDa" verbs. Sohn's approach does not consider these as verbs, but nonetheless, they constitute the majority of all the verbs in Korean. "HaDa" means "do" in English, and always follows a noun. Hence, a noun with "HaDa" attached to it becomes a verb, meaning "to do that noun." One of the typical examples would be "JeonHoaHaDa." Here, "JeonHoa" means telephone in

English. Therefore, "JeonHoaHaDa" means "to call" in English. This verb would have the basic form "JeonHoa." When the usage of this verb has been decided, one of the possible 11 endings would be attached to the end of it. The possible endings are: "HaGiReul," "HaGo IssDa," "HaRa," "HaJa," "Hal GeosIDa," "Hal Ddae," "HanDa," "HanDa," "HanDa," "HanDa," "HanDa." These endings correspond to root form, present continuing, command, request, future, case clause, first singular, second singular, third singular, second plural, and first plural usages in sentences, respectively.

To see how the mechanism works, consider a semantic frame sentence "Call me." This sentence would be recognized as a command sentence. The system looks up the Korean verb mapped to "call," and finds "JeonHoa." Since the verb is categorized as a "HaDa" verb, and since the sentence is recognized as a command sentence, the system searches the ending for command in "HaDa" verbal endings, and finds "HaRa." Then the basic form "JeonHoa" is combined with "HaRa" to make "JeonHoaHaRa."

The regular-consonant-final verbs could be grouped into two classes. The final consonants of these verbs are "S," "D," "B," "T." Although these verbs are classified by Sohn to be linguistically regular, they have not been found to have any apparent relationship with the verbal endings. For example, the words "MudDa" and "BadDa," meaning "to bury" and "to receive" are both D-ending regular-consonant-final verbs, but they belong to two different classes in the current system.

The sole difference between the two classes arises from the ways that command sentences are treated. The first class has an "EoRa" ending whereas the second class has an "ARa" ending. It should be noted that these two classes could be merged to form one class by using "EuRa" in the place of "EoRa" and "ARa." In normal Korean speech, "EoRa" and "ARa" are almost exclusively used to make command sentences with regular-consonant-final verbs. The only time "EuRa" is used is when discussing the Korean language in a linguistics context or by a minority of military personnel. The "EuRa" is a very authoritative and demanding form of command. It sounds peculiar in modern Korean speech. Furthermore, using "EoRa" and "ARa" instead of "EuRa" would not invoke any confusion in any imaginable context. For

these reasons, "EoRa" and "ARa" were chosen, producing two classes.

Unlike the regular-consonant-final verbs, the three irregular-consonant-final verbs had to have individual classifications. The apparent difference between the classes for the regular-consonant-final verbs and the classes for irregular-consonant-final verbs is that some endings of the latter have "S," "D," and "B" consonants in the front. Adding these consonants was needed because of the way the corresponding verbs are written. For example, "to draw" in Korean is "GeusDa," where "Gues" is the stem of the verb. The future form of this verb is "GeuEul GeosIDa." Notice that the "S" in the stem has been omitted. For this reason, the stem is represented by "Geu." Where the ending requires that "S" be in the stem, the ending has its own "S" in the front, like the present continuous form "S Go IssDa." (See Appendix A to see the various endings of these verbs)

The vowel-final verbs have only one class.

3.3.2 Messages

As indicated in Chapter 2, the basic Korean grammar is very different from the English grammar. The order of words in a sentence, the usage of postpositions rather than prepositions, and various verbal endings are the three most pronounced features among the linguistic phenomena of the Korean language. The messages file captures the particular features of the first two linguistic phenomena.

Consider an English sentence "I am going to school now." This sentence would be translated to "I now school to going am" when following the ordering of Korean with English words. In colloquial Korean, however, the same sentence would be translated to "I now school go." Notice that the word ordering is totally different from that of English and that the postposition has been dropped in the colloquial style. This omission does not distort or misconvey the intended meaning of the sentence under normal circumstances, as the speaker and the listener generally know the topic of the conversation, and phenomena that come with speech, such as intonation, help clarify potential confusion arising from the omissions. For this reason, the current setup of messages uses postpositions whenever possible in order to reduce the likelihood of

ambiguities. This setting, however, is to be modified in the future, as it leads to rather stiltifying speech.

The order is recursively formed as topics, predicates, and clauses of a semantic frame get transformed into phrases of the target language according to the grammar templates. Each template consists of a message, and a sequence of words and keywords that describe the message. Messages are semantic frame words that are to be translated to their corresponding target language words. Keywords are the names of categories to which a group of words with common linguistic aspects belong. OBJECT_PRONOUN, for example, is all the pronoun words in the system that can be used as objectives. In the messages file, the messages are the words in the left-most column in lowercase letters. Each message has its describing words and keywords in its row. An example of a message will help illustrate how word order is decided.

Consider the semantic frame word "pass." When this word is transformed to the target language, its describing keywords state that the words that comprise a phrase with "pass" will follow the order of OBJECT_NOUN, ADV_WHEN, TOPIC, ADV_DEGREE, ADV_MAIN, ADV_SOLE, and PREDICATE, with the PREDICATE being "pass." Simply put, the order of the keywords of each message decides the order of target language words associated with the message. As a semantic sentence gets translated into the target language, each word in the semantic sentence is examined at least once. This ensures that the final output will have the correct order as specified by the messages involved.

To see how each of these messages contributes when a semantic frame gets transformed recursively into the target language, consider the English input sentence "CALL SOME ARTILLERY." The sentence is identified to be of type command1. Under this message, the listed keywords in order are OPENING, ID1, TOPIC, PREDICATE, ID2, and CLOSING. Among these keywords, the only one that is relevant to the sentence is PREDICATE as the sentence does not have opening words, identification words, topics, or closing words. Therefore, the entire sentence is a predicate of type command1. The first word of the predicate is "call." Under the message "call," the listed keywords are OBJECT_PRONOUN, TOPIC, ADV_DEGREE,

ADV_MAIN, ADV_SOLE, and PREDICATE. Now, the relevant keywords are topic and predicate, the topic being "some artillery" and the predicate being "call." Since TOPIC comes before PREDICATE, the Korean words for "some artillery" get put before the Korean word for "call." Finally, the topic "some artillery" gets further analyzed for correct order. Note that the keyword TOPIC for "call" has "Eul" following it, indicating that the topic is the object of the sentence whose predicate is "call." This gets attached right after the Korean phrase for "some artillery." Therefore, the final output becomes "some artillery"Eul" call" expressed in English words in the Korean order.

Notice that there is "np-call" below the "call" message. "np" indicates that "call" is used not as a main predicate, but rather as a predicate modifying a noun phrase. Since the example given uses "call" as its main verb, "call" has been used instead of "np-call."

Notice that there are some lower case words inserted between the keywords such as Eun, GeunCheoE, or Eul. Most of these are postpositions. Unlike the keywords which are written in upper case letters and have ":" in front, these words are not linguistic categories, but simply words that later will appear as they are written. They don't always appear, however. Only when the keywords that they follow have nonempty values do they take any values and appear as they are written.

One class of the Korean postpositions is used to indicate that what precedes is a subject, as discussed in detail in Chapter 2: Eun, Neun, I, and Ga. In brief summary, two of them have the same meaning, but are used differently depending on the ending of the subject. If the subject ends with a vowel, the postposition is "Neun." If the subject ends with a consonant, the postposition is "Eun." The messages file has Eun by default. When the subject is found to end with a vowel, then Eun is replaced by Neun. This finding and replacing is done by Rewrite rules which will be discussed in the next section. The other two postpositions are "I" and "Ga" with the same meaning. The difference between these two and the two above is that these two are used for nouns that have definite particle in English. This subclass is not used in the current system setup because the parse tree decoder currently ignores the difference

between nouns with articles and nouns without articles.

Another class of postpositions indicates that what precedes is an object. This class can be divided into two subclasses: one for direct objects, and the other for indirect objects. For now, the assumption is made that all objects are direct objects. This assumption has not caused any problems, since the training sentences do not contain any indirect objects. Furthermore, this assumption simplifies the system setup such that there only needs to be two prepositions for this class: "Eul" and "Reul." "Eul" is used when the ending of the preceding object ends with a consonant, and "Reul" is used when the object ends with a vowel. The default is "Eul"; just as in the case of "Neun" and "Eun," rewrite rules replace this with "Reul" when necessary.

Semantic frame prepositions such as "at," "of," "to," "near," "from" proved to be very troublesome because they have many different meanings and therefore possible translations. Consider two English sentences that use "at": "The plane arrived at 10 AM" and "He pointed at me." "At" means "E" in the first sentence and "EGe" or "Reul" in the second sentence in Korean. TINA and GENESIS have the capability to assign different roles for prepositions, however, depending upon the meaning of the associated noun phrase [7]. This allows having semantically specific prepositions in the lexicon that know precisely which form they should translate to. The current system does not yet fully exploit this feature, however. The remedy used instead is to choose the most general translations. This scheme will soon be changed.

3.3.3 Rewrite rules

There are two columns in rewrite rules. The first column is a list of characters that is searched after, and the second column is a list of characters that will replace the element in the first column once it has been found. There are three subsections to complete the task.

The simplest section deals with postpositions such as "GgaJi," "ESeo," "Geun-CheoE" that were used in the messages table as translations for "up to," "at," and "near," respectively. Rewrite rules replace these postpositions in English characters with those in Korean characters.

The second section completes the verbal classifications of the lexicon. For example, category V4 has "S go IssDa" as the ending for present continuous form. The "S" is supposed to be the ending of the root. However, when running GENESIS, "S" is not recognized as the ending, but as a stand-alone consonant, not attached to any word. This section fixes this problem by eliminating the space between the root and "S." Since there are numerous combinations of this sort, a simple program has been written to automatically generate such combinations with a small set of short tables as its input. This program also automates the last section.

The last section completes the postpositions proposed in the lexicon. As explained, "Eun" is set to be the default postposition for indicating subjects. This is correct only when the ending of the subject is a consonant. When it is not so, this section changes "Eun" to "Neun."

These rewrite rules were found to be very long and largely patterned such that a program could be written to automatically generate the rules. (See Appendix C) The program uses three input data files: "first-consonants," "all-vowels," and "final-consonants." "First-consonants" contains all the consonants that can come in the beginning of a Korean syllable. "All-vowels" lists all Korean vowels, and "final-consonants" lists only the relevant consonants for the rewrite rule generation. The program first generates all the permutations of the three files. These permutations are written in romanized Korean characters. Some of these permutations are not used in Korean at all. To sift out these impossible ones, a program is used to convert the romanized Korean into Korean. During this conversion process, the impossible outcomes are represented by blanks. Then this rough list of Korean syllables gets converted back to romanized Korean. The blanks are removed, producing a clean chart of rewrite rules. Finally, this clean chart gets converted to Korean. Rules computed in this way get combined with a list of rules that specify special cases, ultimately generating the korean-rewrite-rules text file.

In the process of producing the three GENESIS tables necessary for Korean generation, it has been found that GENESIS has some deficiencies for Korean. These deficiencies stem from either the inherent linguistic nature of Korean or the fact that

GENESIS is still an evolving task. The following chapter gives suggestions to improve GENESIS to accommodate some of the deficiencies.

Chapter 4

Proposed improvements to GENESIS

In this chapter, the Korean language phenomena that are not currently being handled adequately by TINA and/or GENESIS are discussed. Note that there are two reasons for this. One is that TINA and GENESIS are still evolving and improving, implying that what cannot be handled at this point are not necessarily due to inadequacies of TINA or GENESIS, but may simply due to lack of necessary mechanisms that have not been implemented yet. Handling negations and passive voice sentences is such an example. The other is that Korean is so different from English that linguistic phenomena occurring in English simply cannot be represented in Korean and vice versa. Translating prepositions to postpositions illustrates this point, for example. For each of the following cases, a suggestion for implementation to solve or reduce the translation problem is given.

4.1 Negations and passive voice sentences

The current generation system handles only a subset of all the possible kinds of Korean sentences, i.e., it does not have a mechanism to handle negation sentences, and it restores passive voiced sentences to active voice. This is a byproduct of the choice of training data, which is a transcription of Task Force Command Net exercise con-

trol messages, chosen to train Common Coalition language at LINColn Laboratory (CCLINC) to suit brigade communications [4]. Since the control messages are usually expressed in positive and active voice, the parser did not have to be concerned with analyzing negation or passive voice sentences, hence the current lack of such a mechanism in the generation system [10].

Once the grammar can analyze such kinds of sentences, the modifications needed for the generation system would be quite simple since negation and passive voice are all reflected and handled solely by postpositions and ending of verbs [8]. Specifically, a sentence could be negated only by changing the ending of the main verb, and an active voice sentence could be transformed into a passive voice sentence by replacing the postpositions for the subject and the object and also changing the ending of the main verb. The following are such examples. Notice that (1-d) is a passive voice negated sentence.

```
(1-a) GoYangIGa JuiReul JabAssDa - positive and active
      CAT       MOUSE   CAUGHT

(1-b) GoYangIGa JuiReul JabJi MosHaissDa - negative and active
      CAT       MOUSE   CATCH DID NOT

(1-c) JuiGa GoYangIEGe JabHyeossDa - positive and passive
      MOUSE CAT        CAUGHT

(1-d) JuiGa GoYangIEGe JabHiJi AnhAssDa - negative and passive
      MOUSE CAT        CATCH   DID NOT
```

Because GENESIS is table-driven, the necessary modifications can be implemented quite easily. The messages file would need to have a message which handles passive voiced sentences, and the lexicon file would need to have extended verbal classifications to accommodate negations and passive voices. Note that having negated sentences may not necessitate making a new message in the messages file as the only deviation from the statement message, which is already implemented in the messages file, is in its verbal ending, handled solely by the lexicon file.

A possible approach that can be taken to incorporate negations and passive voice

would be to use the modal method as used for the auxiliary verb "WILL" in the lexicon. By setting the mode for negations and passive voice, it would become possible to introduce new verbal inflections for each of 8 verbal categories to generate the correct verbal endings. For example, the first category can have a new mode "PASSIVE," followed by "DoiDa" to take care of the passive voice sentences. Careful attention will be needed, however, when this passive voice is accompanied by another mode such as "WILL." In that case, a mechanism that will take multiple modes will be necessary. Similar arguments apply for negations. This approach can be extended to cover the enormous numbers of inflection endings in Korean as follows.

Korean verbal endings usually have more than one inflection. Inflections include passive, honorific, sentence marker, etc. Each of these inflection modes has several variations, and the proper inflection is chosen based both on the verb stems and the two preceding syllables [11]. The inflections also occur in a fixed order as follows [11].

```
Verb stem + Passive + Honorific + Negative + Tense + Sentence marker
```

With the exception of tense and sentence marker, the inflections are optional. Each inflection mode contains more than one variation. Some of the inflections that occur often are listed below.

1. Passive - "Doi," "I," "Hi," "Gi"
2. Honorific - "Si," "EuSi"
3. Negative - "Anh, "JiAnh"
4. Tense (Past) - "Ass," "Eoss," "ss"
5. Tense (Present) - "Eun," "n"
6. Tense (Future) - "Gess"
7. Sentence marker (Declarative) - "Da"
8. Sentence marker (Interogative) - "Ni"
9. Sentence marker (Authoritative) - "Ra"

The inflection modes and the inflection variations listed above are not exhaustive. However, for the purpose of battle management, they are sufficient to generate possibly fluent and adequate verbal endings. The missing modes or inflection variations are either extraneous for our purpose or are still not clearly recognized in the Korean linguistics community [11].

If GENESIS had a capability of defining a ":VERB_MODE" line in the messages file that orders the various modes, and a set of modal settings specified for different kinds of verbs in the lexicon file along with some code modification to attach all those modal endings to the verb stem, then the following idea is proposed to handle the complexity of Korean verb endings.

Let us look at figure 4-1. Figure 4-1 depicts what inflections "HaDa" verbs take and how they should be composed to form a complete ending. Each arrow indicates what inflections can follow a particular inflection. To illustrate the flow, let us examine the word "SiJagHaDa," which means "begin" in English, under circumstances.

1. "Begin" with Past + Declarative
2. "Begin" with Honorific + Present + Interogative

"Begin" with past tense and declarative sentence marker follows the following scheme. The arrow flow is marked with A1 and A2. The arrows begin with the verb stem "SiJag." Then it is attached with "Haiss" to form the past tense inflection. Finally, "SiJagHaiss" becomes combined with "Da" to form the complete verb representing "begin" with past tense and declarative sentence marker. Note that for this process to work, GENESIS has to be able to 1) recognize "SiJag" to be a "HaDa" verb, 2) skip the passive, honorific, and negative modals, 3) recognize "Haiss" to be the correct tense inflection representing past 4) "Da" is the sentence marker for declarative sentences, and 5) to combine them in the correct order.

"Begin" with honorific inflection, present tense, along with interogative sentence marker follows a similar procedure as above, although it is a bit more complicated. Arrows B1, B2, B3 indicate the flow. These arrows, as explained, indicate what

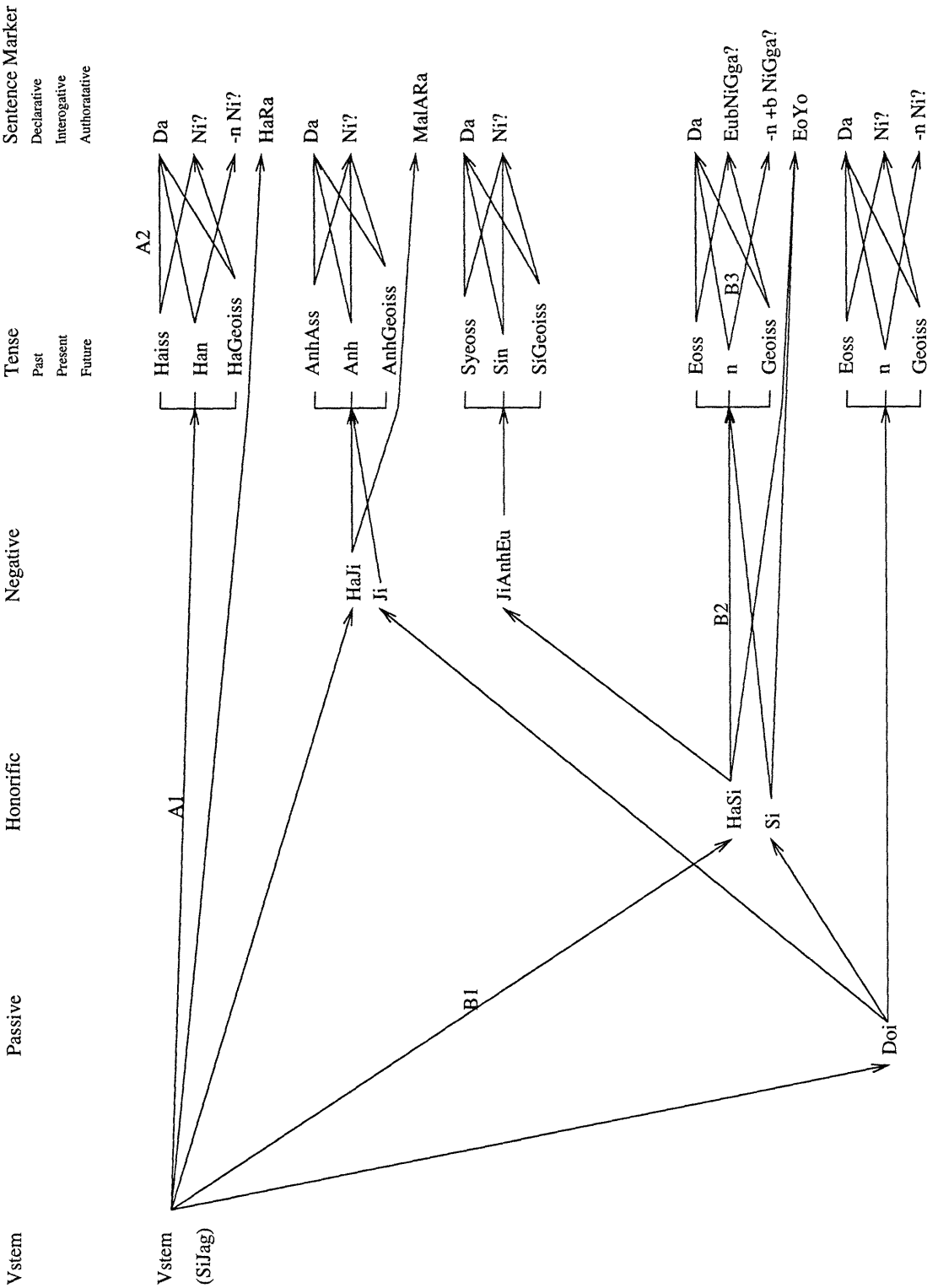

Figure 4-1: Inflections for "Hada" verb

inflections to attach. In this case, they would be "HaSi," "n," "-n +b NiGga." Note that "n" is needed to form present declarative verb ending. However, it is necessary to eliminate this consonant and add "b" to the second syllable of "HaSi." The final form is "SiJagHaSibNiGga." (And the question mark in the end) As can be seen, GENESIS needs to be able to recognize what -n and +b mean in addition to the necessary capabilities mentioned above.

The two instances above illustrate the mechanism of the figure and the necessary capabilities that are needed to be implemented in GENESIS. More or less the verb inflections obey the same procedure described above. Note that "HaDa" verbs belong to the same verb group V as defined in the lexicon file (See Appendix A). There are some exceptions, however. Both "GuHaDa" and "UeonHaDa" do not exactly follow the pattern depicted in figure 4-1. The part that they do not obey is passive inflections. They obey the rest, however. A new verb classification is necessary for this reason. Appendix D shows 5 different sets of inflection patterns. Even though these 5 sets cover a subset of the verbs that the current lexicon file contains, they will serve as a good starting point of generating inflection patterns that would cover the entire spectrum of Korean verbs.

4.2 Articles

The current parser does not exploit its ability to analyze articles [10], but even if it did, articles would not have correct mapping to Korean because Korean does not have exact counterparts to English definite/indefinite articles. Still, if desired, definite articles can be encoded by the Korean demonstratives like "I," "Geu," or "Jeo." Furthermore, indefinite articles can also be encoded by "HaNaEui." Even though these demonstratives can partially capture the meaning of English articles, and therefore would carry more meaning, the resulting translations would sound extremely awkward. Translating articles among different languages is difficult because they do not obey the same linguistic rules.

4.3 Styles of speech

What distinguishes Korean from all other languages is its versatility for expressing the relative positions of the listener and speaker. This includes their ranks, ages, genders, and so forth. Even though these different styles are commonly classified by linguistic terms such as honorific, polite styles, the variety of such styles are so great that a limited number of simple linguistic terms is simply not adequate. For a taste of the variety consider the following example.

```
(1-e)  JeoNeun HagGyoE GabNiDa - an educated child speaking to
       I       SCHOOL  GO        an elderly

(1-f)  JeoNeun HagGyoE GaJiYo  - less formal than (1-e)

(1-g)  JeoNeun HagGyoE GaYo    - less polite than (1-e)

(1-h)  Jeo HagGyoE GaYo        - less formal than (1-g)

(1-i)  NaNeun HagGyoE GaYo     - a child speaking to an older person

(1-j)  Na HagGyoE GaYo         - less formal than (1-i)

(1-k)  Na HagGyoE GanDa        - a friend speaking to a friend

(1-l)  Na HagGyoE Ga           - same as (1-k)

(1-m)  Na HagGyoE GanDa Yai    - female speech of (1-l)

(1-n)  HagGyoE GaJi            - an older person speaking to
                                 a younger person

(1-o)  HagGyoE Ga              - same as (1-n)

(1-p)  HagGyoE GanDanDa        - a female speaking to a younger person

(1-q)  HagGyoE GanDaGuYo

(1-r)  HagGyoE GanDaGu
```

The examples above are far from exhaustive. Although the examples are numerous for a language, by simply switching the verbal classification section of the lexicon file,

it is possible to produce the right kinds of style. This will require an additional discourse module to the existing system to be able to figure out in which context the source language is used.

4.4 Preposition vs. postposition

One of the striking differences between English and Korean is that Korean uses postpositions instead of prepositions. Since postpositions serve similar functions as prepositions, it is usually the case that prepositions get translated into postpositions and vice versa. However, as much as this translation approach seems the only possible choice, this is bound to fail mainly because there are multiple meanings to a single preposition in English. If the meaning of a particular preposition in a sentence can be extracted and represented perfectly in the semantic frame, this problem might be eliminated. However, this is very difficult to achieve. Even if the analysis component does a perfect task of distinguishing each meaning of a particular English preposition, Korean might not even have postpositions that correspond to all the distinguished meanings, hence failing the one-to-one mapping method used in Korean language generation.

Note that this problem is even more severe in transfer method approach. The machine translation systems developed at the Korean Advanced Institute of Science and Technology (KAIST) and at Seoul National University (SNU), suffer from the same problem, as evidenced by the test evaluations documented in MITRE [12]. Their systems replace default Korean postpositions with English prepositions, and this approach often produces incorrect and extremely awkward translations.

In dealing with the issue of prepositions versus postpositions, the interlingua approach has an advantage because each of the various meanings of a particular preposition of English can be mapped to a different semantic meaning representation. If a transfer approach is used, only one semantic meaning can be mapped with each preposition, which often results in incorrect and/or awkward translations.

4.5 Mapping approach

The one-to-one mapping approach without sufficient analysis, and therefore inadequate semantic frames, causes yet another problem. This problem is best illustrated with an example.

Let us consider the English phrase "TRY TO OBTAIN." "TO OBTAIN" corresponds to GuHaGiReul and "TRY" corresponds to SiDoHaDa. Combining these two would be GuHaGiReul SiDoHaDa. However, this is a root and an extra "n" needs to be added to the second to last syllable to make GuHaGiReul SiDoHanDa, which is a present tense verb. Even so, this still sounds awkward because the natural way of saying "TRY TO OBTAIN" is GuHaRyeoHanDa with GuHaRyeoHaDa as its root. Therefore, in order to produce the more natural output, GuHaRyeoHanDa, the semantic frame would have to be complete enough not only to represent the meaning of each word, but the meaning of the phrase that the word belongs to. In addition, the mapping in the language generation would have to contain such cases as well.

Given that the parser is robust enough to identify such verbal phrases, and that the semantic frame can also embrace the meanings of such phrases, the following approach can be taken in modifying the language generation to augment such verbal phrases. This suggested approach is very similar to the approach suggested for handling negations and passive voice sentences, discussed in section 1.3.1 of Chapter 1, i.e., to treat "TRY" as a modal, triggering a particular mechanism that specifies what verbal inflection to use for each of the 8 verbal categories. For the example cited above, the corresponding inflection would be "GyeoHanDa" with "GuHa" as the root of the verb. Just as with the cases for negations and passive voice, a mechanism that would handle multiple inflections will be needed for verbs that have more than one mode. An example would be future "TRY" verbs.

Even with all these modifications, the final output does not sound quite natural. A typical Korean would say the phrase in present continuing tense, "GuHaRyeoNeun JungIDa" which means "IN THE MIDDLE OF TRYING TO OBTAIN." Although "GuHaGiReul SiDoHanDa" for "TRY TO OBTAIN" is not incorrect, it sounds very

textual. "GuHaRyeoNeun JungIDa" would be the most natural translation, which is not the case with the current setup of the system.

4.6 Lexical incompatibility

When translating a language to another language of the same root, it is relatively easy to find equivalents. However, when English is translated into Korean, an English word can have multiple translations in Korean, or it may not have a translation at all. Refer to section 3.3.1 of Chapter 3 for further discussion of this subject.

Chapter 5

Evaluation

5.1 Evaluation Procedure

5.1.1 Data

The transcription of a Task Force Command Net exercise was used to evaluate the performance of the system. Note that this transcription is the same one that was used to train the system. This decision was made because of the following reason; CCLINC is in its infancy and has numerous deficiencies, and the purpose of this document is not to test how well CCLINC can perform, but rather to identify such deficiencies and give suggestions for improvements. Therefore, it was necessary to examine how CCLINC behaved with the training data. The data contain 530 sentences of which 325 sentences are distinctive. The redundant ones are discarded for the purpose of evaluation.

5.1.2 Method

The resulting translations were categorized under two headings: unparsed or parsed. The parsed sentences are evaluated based on how closely the meaning has been preserved (adequacy) and how fluent the translation sounds (fluency). This evaluation was carried out by four native Korean speakers, who scored each translation from 5 to 1, 5 being the best and 1 being the worst. The four scores for each translation

Table 5.1: Evaluation scores

	Adequacy scores	Percentage of all data	Percentage of parsed data	Fluency scores	Percentage of all data	Percentage of parsed data
1	0	0	0	0	0	0
2	1	0.31	0.81	2	0.62	1.61
3	5	1.54	4.03	13	4.00	10.48
4	20	6.15	16.13	29	8.92	23.39
5	98	30.15	79.03	80	24.61	64.52

Table 5.2: Occurrences of each error source

Insufficient analysis of TINA	15
Inadequacies of GENESIS	
-fixable by changing rules	6
-require code modification	6
Other	7

were averaged and rounded to an integer.

5.1.3 Scores

The results are shown in table 5.1, figure 5-1, and figure 5-2. Note that 201 sentences, which contribute 61.85%, failed to be parsed.

5.2 Analysis

Although the majority of the translations for the parsed sentences scored 5 for both adequacy and fluency, a rather large number of parsed sentences resulted in unsatisfactory translations. The causes for the unsatisfactory translations can be from insufficient analysis of input sentences by TINA, or inadequacies of GENESIS for Korean, or other linguistic phenomena that are not related with TINA and GENESIS.

Table 5.2 shows the sources of errors and their distributions. The errors are what causes the translations either inadequate or influent, and the distributions are the

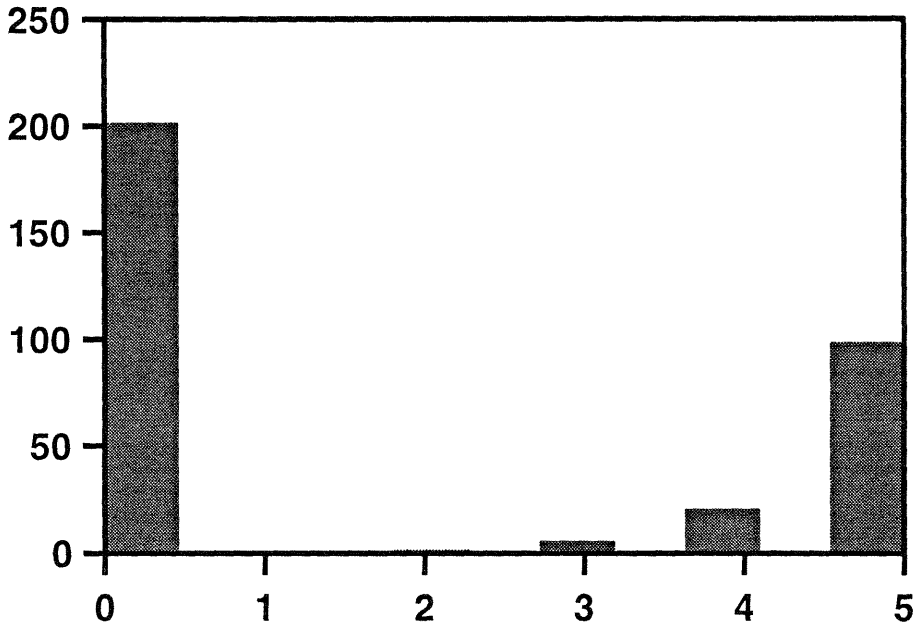

Figure 5-1: Adequacy score histogram (0 indicates unparsed)

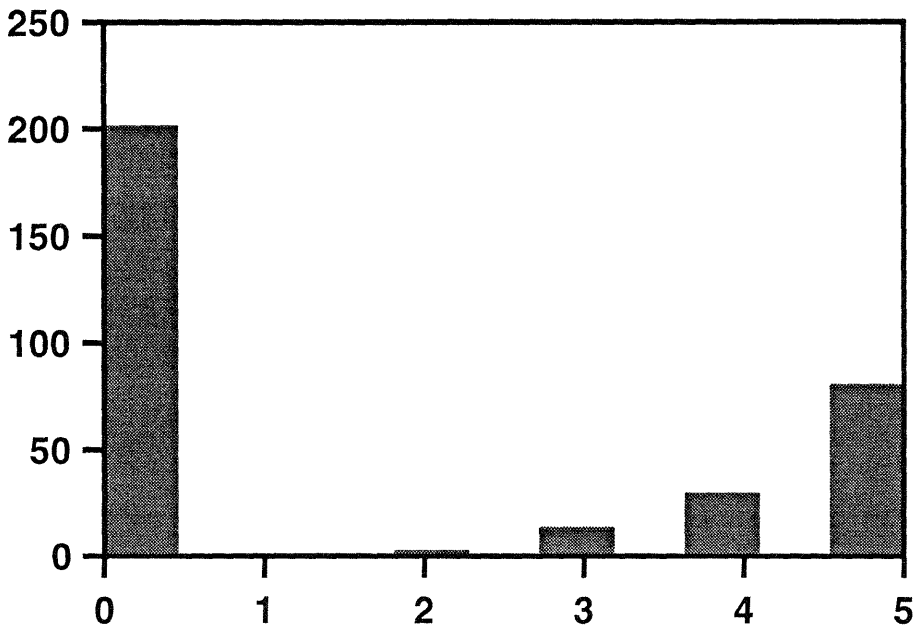

Figure 5-2: Fluency score histogram (0 indicates unparsed)

numbers of events that each error category happens. These numbers are not directly from the scores listed in table 5.1. They are obtained by analyzing the translations with scores less than 5/5, and counting the events that the errors occured. The following sections illustrate problems that are caused by each of the categories in table 5.2.

5.2.1 Insufficient analysis of TINA

```
MOVE BATTALION DELTA TO HILLTOP CHARLIE      - input
ChoalLi GoJiGgaJi DelTa DaiDaiReul UmJigIRa - translation
STIR battalion delta UP TO hilltop charlie    - translation in English
```

There are two problems with the translation and they all stem from insufficient analysis. The problems are in effect one in a sense that they all suffer from lexical ambiguity. "Move," for example, can mean "to go from one point to another," "to change one's residence," or "stir," etc. Also the preposition "to" assumes multiple roles. TINA is certainly capable of distinguishing the different meanings of a particular word. Therefore, fixing this kind of problem would be an easy task.

```
WE ARE OBSERVING THE ENEMY ON THE NORTH AND THE WEST - input
URiNeun Bug HaGo SeoESeo JeogGunEul GoanChalHanDa    - translation
we OBSERVE the enemy on the north and the west       - translation
                                                       in English

MINEFIELD DISCOVERED NEAR SECTOR ALPHA                - input
URiNeun AlPa GuYeog GeunCheoE JiRoiReul BalGyeonHanDa - translation
we DISCOVER minefield near sector alpha               - translation
                                                       in English

WE ARE ENGAGED       - input
URiNeun GoChagHanDa - translation
we ENGAGE            - translation in English
```

The three examples above contain problems caused by ignoring that the sentences are either present continuous tense, past tense, or passive voice. As with the first example, exploiting TINA's capability would resolve this kind of problem.

```
I GOT FOUR BMPS OVER                                          - input
NaNeun SoReyon Gyeong JangGabCha SaReul SoYoHanDa ISang - translation
i POSSESS four bmps over                                      - translation
                                                                in English
```

Again, the word "got" has a multiple meaning, and the analysis failed to pick the correct meaning.

```
SEND AGAIN    - input
BanBogHaRa    - translation
REPEAT        - translation in English
```

TINA parsed this input to mean "repeat." Because of this incorrect parse, the translation is also incorrect.

5.2.2 Fixable by changing rules of GENESIS

```
REQUEST PERMISSION TO DEFEND HILLTOP ECHO - input
URiNeun EKoGoJiReul ChugSeongHaGiReul HeoGaReul YoGuHanDa
                                          - translation
```

Redundant usage of the postposition "Reul" makes a translation that could be fluent otherwise. Instead of putting postpositions every time there is an object in the messages file, put them at places where they are absolutely necessary.

```
OH WAIT     - input
A GiDaRiRa - translation
```

The problem with this translation was pointed out by a grader. Authoritative commands in Korean can be classified into two categories. One can be said to have either "EoRa" "YeoRa"ending whereas the other one usually has an "ARa," "EuRa" or "IRa" ending. The former inflection is used by most people including civilians and off-duty military personnel. It is considered to be standard inflection for authoritative command sentences. "ARa" and "EuRa" are almost never used by civilians in normal

conversations or writing. If used at all, it would be by military personnel. However, one of the graders commented that the latter inflections are rarely used now in Korea, and they would sound awkward even to military personnel. Therefore, the more natural translation would be "A GiDaRyeoRa," and this can be easily fixed in the lexicon file.

```
AFFIRMATIVE      - input
DanJeongJeogIDa - translation
```

Although "DanJeongJeogIDa" is not an incorrect translation, "GeuReohDa" would be a better translation as it is more widely used.

5.2.3 Require code modification for GENESIS

One of the most difficult problem with the Korean language generation deals with choosing the right inflection endings for verbs. The following example illustrates this point.

```
I AM TRYING TO GET A GRID NOW - input
NaNeun JoiPyoReul JiGeum GuHaGiReul SiDoHanDa - translation
```

The problem with this translation occurs because GENESIS tries to map "trying to" and "get" with two different words whereas the natural translation uses one verb for "get" with an inflection ending that incorporates the meaning of "trying to." Refer to section 4.5 of Chapter 4 to see the discussion in depth.

5.2.4 Other

In this subsection, the discussion focuses on the problems that occur not because of inadequacies of TINA or GENESIS, but because of the greatly different linguistic natures of English and Korean. These problems propose the greatest difficulty in translating Korean from English.

```
LEAD ELEMENTS OF MY UNIT NOW PASSING PHASE LINE ALPHA - input
Nai BuDaiEui SeonDuBuDaiNeun JiGeum AlPa TongGyeSeonEul TongGoaHanDa
                                                    - translation
```

This example shows the cultural difference reflected in the languages. What can be considered to belong to a person in English is often thought to belong to a group in Korean. Although the translation is both adequate and fluent, the more natural translation would use "URi," meaning "our," instead of "Nai," meaning "my."

```
FIRST BATTALION COMMANDER REPORT YOUR LOCATION     - input
CheossJjai DaiDaiEui BuDaiJang Ne JangSo BoGoHaRa - translation
```

It is natural to use a cardinal number in expressions such as "first battalion." However, for such an expression, Koreans use "three battalion" instead.

```
WE ARE NOW GOING TO GET INTO THEIR MAIN DEFENSIVE BELT - input
UriNeun JiGeum GeuDeulEui JuYoHan BangEoYoDaiReul ChimTuHaGiReul GanDa
                                                    - translation
```

Contracted forms are used very frequently in Korean. Although using "JuYoHan BangEoYoDai" is both adequate and fluent, using "JuBangEoYoDai" for "main defensive belt" sounds even more fluent.

```
ONE BMP AND ONE SAGGER TEAM OVER - input
SoRyeon Gyeong JangGabCha Il HaGo SaGaTim Il ISang - translation
```

As discussed in section 3.3.1 of Chapter 3, there are two ways of reading arabic numbers. When numbers are used to count items as in this case, pure Korean is used. Therefore, "HaNa" should be used instead of "Il" for "one."

5.3 Conclusion

The scores on the translations of the test sentences indicate that nearly 80% of the parsed sentences have reasonable adequacy and nearly 65% of the parsed sentences

have acceptable fluency. Most of the problems that contribute the rest of the parsed sentences arise from either under-utilizing the capabilities of TINA and GENESIS or their infancy stage. With improved rules and augmented codes for TINA and GENESIS, the future evaluation is believed to result in better scores. The problems discussed in section 5.2.4, however, propose series difficulty in the translation and require further research.

Chapter 6

Discussion and future plans

The degree to which GENESIS is able to handle Korean language generation in an interlingua system has been described in this thesis. The system has been trained with and tested on a transcription of a Task Force Command Net exercise. The two measures of evaluation, adequacy and fluency, indicate that nearly 80% of the parsed sentences are reasonably good translations in the sense that they carry the correct meaning of the original sentences, and that approximately 65% of the parsed sentences sound natural to native Korean speakers.

The current system is, however, an evolving system. The internal engines of TINA and GENESIS are constantly improved to handle more complex and new sentences. The grammar rules for TINA are being developed further to accommodate the linguistic phenomena that cannot be handled by current rules, such as negations, passive voice, and articles. Along with these improved rules, a more exhaustive semantic frame is being developed. This more exhaustive semantic frame would resolve the lexical ambiguities of the source language.

Given the improved setup for TINA and the new semantic frame, better parsing can be expected, making correct language generation a more feasible task. Certainly, given the right parses, the adequacy measure can be expected to improve drastically, as even a string of correct Korean equivalents to the English input would allow one to extract the intended meanings of the input sentences. However, improving the other measure, fluency, is believed to be a more difficult task.

Even when parsing has been done correctly, generating Korean translation by putting the right nouns, adequate postpositions, and verbs with appropriate inflections might produce very awkward output. The awkwardness can happen due to several reasons. One obvious reason is that English idiomatic expressions may produce totally unrelated strings of Korean words when translated in the way described. Another one is that a natural Korean expression might employ a set of words for which an equivalent English expression does not exist. For example, the natural Korean translation for "can you buy it for me?" is "can you buy and give it to me?" when translated back to English. Because of dissimilarities such as these between the two languages, achieving fluent Korean translation is believed to be a hard task.

The evaluation process will also need to be augmented. One of the tendencies that has been noticed when evaluating some preliminary translations is that the evaluators become used to the translation patterns so that they unconsciously start to believe that the translations were more correct as the evaluation progressed. To prevent this from occurring, evaluators would need to be divided into two groups: one group would be provided translations on paper, and the other group would listen to a Korean speech synthesizer for evaluation.

A Korean speech synthesizer named "Says," produced by Digicom (in Korea) was acquired for this purpose, but has not been completely installed due to a software component which is lacking at this moment. When it is incorporated into the system, the evaluation procedure outlined above will be possible.

Appendix A

Lexicon for GENESIS

```
6                D "Yug" CARDINAL "YeoSeosJjai"
7                D "Chil" CARDINAL "IlGobJjai"
8                D "Pal" CARDINAL "YeoDeolbJjai"
9                D "Gu" CARDINAL "AHobJjai"


N                N " "
a-4              N "a-4"
a-6              N "a-6"
a-7              N "a-7"
a-10             N "a-10"
air              N "HangGong"
air_alert        N "GongSeubGyeongBo"
air_combat_fighter  N "JeonTuGi"
air_strike       N "DaiGongGongGyeog"
air_support      N "HangGongJiUeon"
airplane         N "BiHaingGi"
alligator        N "AgEo"
aloc             N "HangGong ByeongChamSeon"
alpha            N "AlPa"
alpha_bravo      N "AlPa BeuRaBo"
ammo_status      N "TanYag SangTai"
artillery        N "PoByeong"
attack           N "GongGyeog"
attention        N "JuEui"
battalion        N "DaiDai"
bear             N "Gom"
belt             N "YoDai"
bmp              N "SoRyeon Gyeong JangGabCha"
bmp_team         N "SoRyeon Gyeong JangGabCha Pyeon"
bravo            N "BeuRaBo"
bridge           N "GyoRyang"
bridge_report    N "GyoRyang BoGo"
charlie          N "ChoalRi"
checkpoint       N "GeomMunSo"
cheetah          N "ChiTa"
commander        N "BuDaiJang"
company          N "JungDai"
contact          N "JeobChog"
coordinated_attack  N "HyeobDongGongGyeog"
corsair          N "HaiJeogSeon"
crocodile        N "AgEo"
```

Table A.1: Lexicon file for GENESIS

```
A               A "HaNaEui"
close           A "GaGgaUn"
defensive       A "BangEo"
front           A "ApEui"
left            A "OinJjog"
right           A "OReunJjog"
heavy           A "DaiGyuMoEui"
1               A "Il" CARDINAL "CheosJjai"
quick           A "BbaReun"
rough           A "GeoChilEun"
main            A "JuYoHan"
this_is         A "YeoGiNeun"
unknown         A "AlRyeoJiJi AnhAxDa"

at_time         C " "
and             C "HaGo"
or              C "INa"

are             X " "
command1        CL "CL" MODE "imp1"
command2        CL "CL" MODE "imp2"
is              X " "
to              X " " MODE "root"
when            CL " " MODE "case"
will            X " " MODE "future"

def             D " "
first           D "CheosJjai"
indef           D " "
my              D "Nai"
no_det          D " "
some            D "JoGeum"
your            D "Ne"
his             D "GeuEui"
our             D "URiEui"
their           D "GeuDeulEui"
0               D "Yeong"
2               D "I" CARDINAL "DulJjai"
3               D "Sam" CARDINAL "SesJjai"
4               D "Sa" CARDINAL "NesJjai"
5               D "O" CARDINAL "DaSeosJjai"
```

```
defensive_belt  N "BangEoYoDai"
delta           N "DelTa"
delta_charlie   N "DelTa ChoalRi"
digger          N "GaingBu"
dismount        N "NagChaGun"
dragon          N "Yong"
eagle           N "DogSuRi"
east            N "Dong"
echo            N "EKo"
element         N "YoSo"
enemy           N "JeogGun"
eta             N "YeSangDoChagSiGan"
forces_motorized  N "GiDongByeongRyeog"
foxtrot         N "PogSeuTeuRosTeu"
fran            N "PeuRain"
ghostrider      N "YuRyeongGiSa"
grid            N "JoaPyo"
gunners         N "SaSu"
hawk            N "Mai"
hilltop         N "GoJi"
hotel           N "HoTel"
id              N " "
infantry        N "BoByeong"
intruder        N "ChimIbJa"
juliet          N "JyulRiEs"
laying          N "SeolChi"
lead_element    N "SeonDuBuDai"
leopard         N "PyoBeom"
lieutenant      N "JungUi"
line            N "Seon"
lion            N "SaJa"
location        N "JangSo"
mine            N "JiRoi"
minefield_laying_report  N "JiRoiBat SeolChi BoGo"
motorized_forces  N "GiDongByeongRyeog"
nbc_alert       N "HoaSaingBang GyeongGo"
north           N "Bug"
november        N "NoBemBeo"
object          N " "
objective       N "MogPyo"
op              N "OPi"
operation       N "JagJeon"
```

```
overwatch       N "GamSi"
panther         N "PyoBeom"
passability     N "TongGoaSeong"
permission      N "HeoGa"
phase_line      N "TongJeSeon"
platoon         N "SoDai"
positive        N "GeuReohDa"
ranger          N "YuGyeogByeong"
rear            N "Dui"
resistance      N "JeoHang"
rhino           N "KoBbulSo"
ridge           N "SanMaRu"
road            N "Gil"
saber           N "GiByeongDai"
sagger          N "SaGa"
sagger_team     N "SaGaTim"
scorpion        N "JeonGal"
sector          N "GuYeog"
shark           N "SangEo"
sitrep          N "SangHoangBoGo"
snake           N "Baim"
soldier         N "GunIn"
south           N "Nam"
t-72            N "t-72"
tank            N "TaingKeu"
team            N "Pyeon"
terrain         N "JiYeog"
that            N "JeoGeos"
this            N "IGeos"
tiger           N "HoRangI"
toc             N "JeonSul JagJeonBonBu"
troops          N "GiGab JungDai"
unit            N "BuDai"
vulture         N "DogSuRi"
west            N "Seo"
wolf            N "NeugDai"

P               N "P" G "m"
he              PN "Geu" NUM "third"
him             PN "Geu" NUM "third"
her             PN "GeuNyeo" NUM "third"
i               PN "Na" NUM "first"
```

```
it_obj          PN "GeuGeos" NUM "third"
it_subj         PN "GeuGeos" NUM "third"
people          PN "SaRamDeul" NUM "third"
pro             PN "Na" SECOND "Neo" FPL "URi" SECOND "NeoHeui"
we              PN "URi" NUM "fpl"
she             PN "GeuNyeo" NUM "third"
them            PN "GeuDeul" NUM "pl"
they            PN "GeuDeul" NUM "pl"
you_obj         PN "Neo" NUM "second"
you_subj        PN "Neo" NUM "second"

affirmative     0 "DanJeongJeogIDa"
at_this_time    0 "JiGeum ISiGan"
break           0 "JungJiHanDa"
copy            0 "AlAxDa"
heavily         0 "GyeogRyeolHaGe"
negative        0 "ANiDa"
no              0 "ANiDa"
now             0 "JiGeum"
oh              0 "A"
ok              0 "JohA"
okay            0 "JohA"
over            0 "ISang"
pretty          0 "SangDangHi"
quickly         0 "BbaReuGeo"
roger           0 "AlAxDa"
roger_that      0 "AlAxDa"
yea             0 "GeuReohDa"
yes             0 "GeuReohDa"

V               V "V" ROOT "HaGiReul" ING "HaGo IxDa" IMP1 "HaRa"
IMP2 "HaJa" FUTURE "Hal GeosIDa" CASE "Hal Ddai" FIRST "HanDa"
SECOND "HanDa" THIRD "HanDa" PL "HanDa" FPL "HanDa"

V2             V2 "V2"  ROOT "GiReul" ING "Go IxDa" IMP1 "EuRa"
IMP2 "Ja" FUTURE "Eul GeosIDa" CASE "Eul Ddai" FIRST "NeunDa"
SECOND "NeunDa" THIRD "NeunDa" PL "NeunDa" FPL "NeunDa"

V3             V3 "V3"  ROOT "GiReul" ING "Go IxDa" IMP1 "ARa"
IMP2 "Ja" FUTURE "Eul GeosIDa" CASE "Eul Ddai" FIRST "NeunDa"
SECOND "NeunDa" THIRD "NeunDa" PL "NeunDa" FPL "NeunDa"
```

```
V4              V4 "V4"  ROOT " SGiReul" ING " SGo IxDa" IMP1
"EuRa" IMP2 " SJa" FUTURE "Eul GeosIDa" CASE "Eul Ddai" FIRST
" SNeunDa" SECOND " SNeunDa" THIRD " SNeunDa" PL " SNeunDa"
FPL " SNeunDa"

V5              V5 "V5"  ROOT " DGiReul" ING " DGo IxDa" IMP1
" REoRa" IMP2 " DJa" FUTURE " REul GeosIDa" CASE " REul Ddai"
FIRST " DNeunDa" SECOND " DNeunDa" THIRD " DNeunDa" PL
" DNeunDa" FPL " DNeunDa"

V6              V6 "V6"  ROOT " BGiReul" ING " BGo IxDa" IMP1
"UeoRa" IMP2 " BJa" FUTURE "Eul GeosIDa" CASE "Eul Ddai"
FIRST " BNeunDa" SECOND " BNeunDa" THIRD " BNeunDa" PL
" BNeunDa" FPL " BNeunDa"

V10             V10 "V10" ROOT "GiReul" ING "Go IxDa" IMP1 "Ra"
IMP2 "Ja" FUTURE " R GeosIDa" CASE " R Ddai" FIRST " NDa"
SECOND " NDa" THIRD " NDa" PL " NDa" FPL " NDa"

V11             V11 "V11" ROOT "GiReul" ING "Go IxDa" IMP1 "Ra"
IMP2 "Ja" FUTURE " R GeosIDa" CASE " R Ddai" FIRST " NDa"
SECOND " NDa" THIRD " NDa" PL " NDa" FPL " NDa"

approach        V11 "DaGaGa"
be              V10 " "
begin           V "SiJag"
call            V11 "BuReu"
cross           V11 "GeonNeo"
destroy         V "PaGoi"
discover        V "BalGyeon"
encounter       V10 "ManNa"
engage          V "GoChag"
engaged_with    V "GoChag"
fortify         V "ChugSeong"
go              V11 "Ga"
leave           V11 "DdeoNa"
monitor         V "GamCheong"
move            V10 "UmJigI"
observe         V "GoanChal"
obtain          V "Gu"
pass            V "TongGoa"
pay             V "JiBul"
```

```
pay_attention    V "JuEui"
penetrate        V "ChimTu"
possess          V "SoYu"
receive          V3 "Bad"
repeat           V "BanBog"
report           V "BoGo"
v_request        V "YoGu"
sight            V "MogGyeog"
signal           V "SinHo"
take_action      V3 "Mat"
take_over        V3 "InGyeBad"
think            V "SaingGag"
try              V "SiDo"
use              V "SaYong"
wait             V11 "GiDaRi"
want             V "Ueon"
wave             V10 "HeunDeul"
laugh            V2 "Us"
bury             V2 "Mud"
bend             V2 "Gub"
draw             V4 "Geu"
ask              V5 "Mu"
roast            V6 "Gu"
```

Appendix B

Messages for GENESIS

Table B.1: Messages file for GENESIS

```
command1                :OPENING :ID1 :TOPIC :PREDICATE :ID2 :CLOSING
command2                :OPENING :ID1 :TOPIC :PREDICATE :ID2 :CLOSING
statement               :OPENING :ID1 (:TOPIC pro) *Eun :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE :ID2 :CLOSING
call_up                 :OPENING :ID1 :TOPIC :PREDICATE :CLOSING
reply                   :OPENING :TOPIC :CVC2_MSG :PREDICATE :CLOSING
topic                   :QUANTIFIER :COMPLEMENT :NOUN_PHRASE
np-and                  :NOUN_PHRASE :TOPIC
and                     :NOUN_PHRASE :TOPIC
np-or                   :NOUN_PHRASE :TOPIC

conjunction             :TOPIC1 :CONJUNCTION :TOPIC2

near                    :TOPIC GeunCheoE
np-near                 :TOPIC GeunCheoE :NOUN_PHRASE

np-of                   :TOPIC Eui :NOUN_PHRASE
of                      :TOPIC

np-adj_intensity        :TOPIC :NOUN_PHRASE
adj_intensity           :TOPIC

np-directional          :TOPIC :NOUN_PHRASE
directional             :TOPIC

at                      :TOPIC ESeo
np-at                   :TOPIC ESeo :NOUN_PHRASE

np-degree               :TOPIC :NOUN_PHRASE
degree                  :TOPIC

np-up_to                :TOPIC GgaJi :NOUN_PHRASE
up_to                   :TOPIC

np-from                 :TOPIC ESeo :NOUN_PHRASE :PREDICATE
from                    :TOPIC :PREDICATE

np-to                   :TOPIC :PREDICATE :NOUN_PHRASE
to                      :TOPIC :PREDICATE

this_is                 :PREDICATE :TOPIC :ID2
```

```
begin                   :OBJECT_PRONOUN :ADV_WHEN :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE :TOPIC

take_action             :OBJECT_PRONOUN :TOPIC *Eul :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE

take_over               :OBJECT_PRONOUN :TOPIC :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
np-take_over            :NOUN_PHRASE :OBJECT_PRONOUN :TOPIC
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

pass                    :OBJECT_PRONOUN :ADV_WHEN :TOPIC *Eul
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE
np-pass                 :NOUN_PHRASE :OBJECT_PRONOUN :AUX :TOPIC *Eul
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

pay_attention           :OBJECT_PRONOUN :TOPIC :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
np-pay_attention        :NOUN_PHRASE :OBJECT_PRONOUN :TOPIC
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

phone                   :OBJECT_PRONOUN :ADV_WHEN :TOPIC *Eul
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE
np-phone                :NOUN_PHRASE :OBJECT_PRONOUN :AUX :TOPIC *Eul
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

report                  :OBJECT_PRONOUN :ADV_CLAUSE :TOPIC
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

go                      :OBJECT_PRONOUN :COMPLEMENT :ADV_WHEN
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

cross                   :OBJECT_PRONOUN :TOPIC *Eul :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
np-cross                :NOUN_PHRASE :OBJECT_PRONOUN :TOPIC *Eul
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

when                    :OBJECT_PRONOUN :TOPIC :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE

sight                   :OBJECT_PRONOUN :TOPIC *Eul :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
```

```
observe                 :OBJECT_PRONOUN :TOPIC *Eul :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE

try                     :OBJECT_NOUN :ADV_WHEN :COMPLEMENT
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

obtain                  :OBJECT_PRONOUN :TOPIC *Eul :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
np-obtain               :NOUN_PHRASE :OBJECT_PRONOUN :TOPIC *Eul
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

request                 :OBJECT_PRONOUN :TOPIC *Eul :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
np-request              :NOUN_PHRASE :OBJECT_PRONOUN :TOPIC *Eul
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

move                    :OBJECT_PRONOUN :TOPIC *Eul :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
np-move                 :NOUN_PHRASE :OBJECT_PRONOUN :TOPIC *Eul
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

call                    :OBJECT_PRONOUN :TOPIC *Eul :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
np-call                 :NOUN_PHRASE :OBJECT_PRONOUN :TOPIC *Eul
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

leave                   :OBJECT_PRONOUN :ADV_WHEN :TOPIC
                        :ADV_DEGEE :ADV_MAIN :ADV_SOLE :PREDICATE

encounter               :OBJECT_PRONOUN :TOPIC *Eul :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
np-encounter            :NOUN_PHRASE :OBJECT_PRONOUN :TOPIC *Eul
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

discover                :OBJECT_PRONOUN :TOPIC *Eul :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE

np-destroy              :OBJECT_PRONOUN :TOPIC *Eul :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE

destroy                 :NOUN_PHRASE :OBJECT_PRONOUN :TOPIC *Eul
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE
```

```
fortify                 :OBJECT_PRONOUN :AUX :TOPIC *Eul :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
np-fortify              :NOUN_PHRASE :OBJECT_PRONOUN :AUX :TOPIC *Eul
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

approach                :OBJECT_PRONOUN :TOPIC *Eul :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE

engage                  :OBJECT_PRONOUN :TOPIC :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
engaged_with            :OBJECT_PRONOUN :TOPIC :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE

penetrate               :OBJECT_PRONOUN :TOPIC *Eul :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
np-penetrate            :NOUN_PHRASE :OBJECT_PRONOUN :AUX
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE
                        :TOPIC *Eul

possess                 :OBJECT_PRONOUN :TOPIC *Eul :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
np-possess              :NOUN_PHRASE :OBJECT_PRONOUN :TOPIC *Eul
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

receive                 :OBJECT_PRONOUN :TOPIC :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
np-receive              :NOUN_PHRASE :OBJECT_PRONOUN :TOPIC
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

repeat                  :OBJECT_PRONOUN :TOPIC :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
np-repeat               :NOUN_PHRASE :OBJECT_PRONOUN :TOPIC
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

v_request               :OBJECT_PRONOUN :TOPIC *Eul :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
np-v_request            :NOUN_PHRASE :OBJECT_PRONOUN :TOPIC *Eul
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE
```

```
use                     :OBJECT_PRONOUN :TOPIC *Eul :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
np-use                  :NOUN_PHRASE :OBJECT_PRONOUN :TOPIC *Eul
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

wait                    :OBJECT_PRONOUN :TOPIC :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
np-wait                 :NOUN_PHRASE :OBJECT_PRONOUN :TOPIC
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

want                    :OBJECT_PRONOUN :ID1 *Eul :TOPIC :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
np-want                 :NOUN_PHRASE :OBJECT_PRONOUN :ID1 *Eul
                        :TOPIC :ADV_DEGREE :ADV_MAIN :ADV_SOLE
                        :PREDICATE

wave                    :OBJECT_PRONOUN :TOPIC :ADV_DEGREE
                        :ADV_MAIN :ADV_SOLE :PREDICATE
np-wave                 :NOUN_PHRASE :OBJECT_PRONOUN :TOPIC
                        :ADV_DEGREE :ADV_MAIN :ADV_SOLE :PREDICATE

terrain_type            :TOPIC
np-terrain_type         :TOPIC :NOUN_PHRASE

support_type            :TOPIC
np-support_type         :NOUN_PHRASE :TOPIC

kind                    :TOPIC
np-kind                 :TOPIC :NOUN_PHRASE

unknown                 :TOPIC :ADV_WHEN :ADV_SOLE :PREDICATE

np-at_time              :TOPIC :PREDICATE :NOUN_PHRASE
np-mil_time             :TOPIC SiE :NOUN_PHRASE

;; strange ordering is needed since the order of predicates at the
;; same level is determined by their relative order in this file
code1                   :NOUN_PHRASE :TOPIC
np-code1                :NOUN_PHRASE :TOPIC
code4                   :NOUN_PHRASE :TOPIC
np-code4                :NOUN_PHRASE :TOPIC
digit_code1             :TOPIC
```

```
np-digit_code1          :NOUN_PHRASE :TOPIC
digit_code2             :TOPIC
np-digit_code2          :NOUN_PHRASE :TOPIC
code2                   :NOUN_PHRASE :TOPIC
np-code2                :NOUN_PHRASE :TOPIC
code3                   :NOUN_PHRASE :TOPIC
np-code3                :NOUN_PHRASE :TOPIC
code5                   :NOUN_PHRASE :TOPIC
np-code5                :NOUN_PHRASE :TOPIC
code6                   :NOUN_PHRASE :TOPIC
np-code6                :NOUN_PHRASE :TOPIC
code7                   :NOUN_PHRASE :TOPIC
np-code7                :NOUN_PHRASE :TOPIC
code8                   :NOUN_PHRASE :TOPIC
np-code8                :NOUN_PHRASE :TOPIC
code9                   :NOUN_PHRASE :TOPIC
np-code9                :NOUN_PHRASE :TOPIC
code10                  :NOUN_PHRASE :TOPIC
np-code10               :NOUN_PHRASE :TOPIC
code11                  :NOUN_PHRASE :TOPIC
np-code11               :NOUN_PHRASE :TOPIC

digit_code3             :TOPIC
np-digit_code3          :NOUN_PHRASE :TOPIC
digit_code4             :TOPIC
np-digit_code4          :NOUN_PHRASE :TOPIC
digit_code5             :TOPIC
np-digit_code5          :NOUN_PHRASE :TOPIC
digit_code6             :TOPIC
np-digit_code6          :NOUN_PHRASE :TOPIC
digit_code7             :TOPIC
np-digit_code7          :NOUN_PHRASE :TOPIC
digit_code8             :TOPIC
np-digit_code8          :NOUN_PHRASE :TOPIC

distance                :TOPIC
np-distance             :TOPIC :NOUN_PHRASE

np-cardinal             :TOPIC :NOUN_PHRASE

unit_number             Je :TOPIC :NAME
```

```
np-unit_number          Je :TOPIC :NAME :NOUN_PHRASE
numeric                 :TOPIC
np-numeric              :NOUN_PHRASE :TOPIC

np-nonprec_initials     :TOPIC :NOUN_PHRASE
np-prec_initials        :TOPIC :NOUN_PHRASE
```

Appendix C

Rewrite-rules for GENESIS

Table C.1: Data files used for rewrite.c

first-consonants	all-vowels	final-consonants
G	a	s
N	ya	b
D	eo	l
R	yeo	d
M	o	n
B	yo	
S	u	
	yu	
J	eu	
Ch	i	
K	ai	
T	yai	
P	e	
H	ye	
GG	oa	
SS	oi	
DD	ui	
BB	ueo	
JJ	eui	
	oai	
	ue	

Table C.2: Program automatically generating korean-rewrite-rules.text

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define  MAXLENGTH 10    /* maximum length of character strings */
#define  MAXDATA   100   /* maximum number of vowels or consonants */

main()
{
  FILE     *fp1;          /* pointer to first-consonants file */
  FILE     *fp2;          /* pointer to all-vowels file */
  FILE     *fp3;          /* pointer to final-consonants file */
  FILE     *fopen();
  FILE     *result;       /* pointer to chart file containing */
                          /* the relevant romanized combinations of */
                          /* Korean syllables */
  FILE     *messy;        /* pointer to rough-romanized-chart */
                          /* containing the possible Korean */
                          /* syllables and the impossible Korean */
                          /* syllables represented by quoted blanks */
  FILE     *clean;        /* pointer to romanized-chart containing */
                          /* only the possible Korean syllables */
  char     *firstcon[MAXDATA];  /* first consonants */
  char     *vowel[MAXDATA];     /* all the vowels */
  char     *finalcon[MAXDATA];  /* final consonants */
  int      count1=0, count2=0, count3=0, counter=0, i, j, k, l;
  char     unit[MAXLENGTH];     /* quote + first consonant + vowel */
  char     unit2[MAXLENGTH];    /* final consonant + quote */
  char     unit3[MAXLENGTH];    /* quote + first consonant + vowel */
                                /* + final consonant + quote */
  char     testing[MAXLENGTH] = "\"\"";         /* empty quotes */
  char     testing2[MAXLENGTH] = "\"Reul\"";    /* "Reul" */
  char     testing3[MAXLENGTH] = "\"Neun\"";    /* "Neun" */
```

```
/* ================================================================ */
/* Open all the necessary files to find the possible combinations  */
/* These files include first-consonant, all-vowels, and            */
/* final-consonant                                                 */
/* The consonants and vowels are in romanized form                 */
/* ================================================================ */

  if ((fp1 = fopen("first-consonants", "r")) == NULL){
    printf("Cannot open first-consonants file.\n");
    exit(1);
  }
  else
    {
      while (fscanf(fp1,"%s",&firstcon[count1]) != EOF){
        count1++;
        firstcon[count1]=" ";
      }
      fclose(fp1);
    }

 if ((fp2 = fopen("all-vowels", "r")) == NULL){
    printf("Cannot open all-vowels file.\n");
    exit(1);
  }
  else
    {
      while (fscanf(fp2,"%s",&vowel[count2]) != EOF)
        count2++;
      fclose(fp2);
    }

 if ((fp3 = fopen("final-consonants", "r")) == NULL){
    printf("Cannot open final-consonants file.\n");
    exit(1);

  }
  else
    {
      while (fscanf(fp3,"%s",&finalcon[count3]) != EOF)
        count3++;
      fclose(fp3);
    }
```

```
/* ============================================================ */
/* The Korean syllables get produced by permutating             */
/* first-consonants, all-vowels, and final-consonants           */
/* If a syllable does not have a final-consonant, the attached  */
/* "Eul" becomes "Reul," and "Eun" becomes "Neun"               */
/* ============================================================ */

  if ((result = fopen("chart", "w")) == NULL){
    printf("Cannot open chart file.\n");
    exit(1);
  }
  else
    {
      fprintf(result,"\\begin{sshr}\n\n");
      /* this is necessary to run sshr2ks */

      for(k=0;k<count3;k++)
        for(j=0;j<count2;j++)
          for(i=0;i<=count1;i++) {
            fprintf(result,"\"%s%s %s\"                \"%s%s%s\"\n",
                    &firstcon[i],&vowel[j],&finalcon[k],
                    &firstcon[i],&vowel[j],&finalcon[k]);
            counter++;}
      for(j=0;j<count2;j++)
        for(i=0;i<=count1;i++)
          fprintf(result,"\"%s%sEul\"                \"%s%sReul\"\n",
                  &firstcon[i],&vowel[j],&firstcon[i],&vowel[j]);
      for(j=0;j<count2;j++)
        for(i=0;i<=count1;i++)
          fprintf(result,"\"%s%sEun\"                \"%s%sNeun\"\n",
                  &firstcon[i],&vowel[j],&firstcon[i],&vowel[j]);


      fprintf(result,"\\end{sshr}");


      system("sshr2ks chart > rough-korean-chart");
      /* converts romanized-Korean to Korean characters */
      /* in this process, the impossible ones become blanks */
```

```
      system("ks2sshr rough-korean-chart > rough-romanized-chart");
      /* converting back to rough-korean-chart to do some cleaning */
      /* up of these impossible ones */

      fclose(result);
    }

/* ============================================================== */
/* The following eliminates the impossible syllables              */
/* They are the entries that contain either blanks, "Eul," or     */
/* "Reul"                                                         */
/* ============================================================== */

  if ((messy = fopen("rough-romanized-chart", "r")) == NULL){
    printf("Cannot open messy-chart file.\n");
    exit(1);
  }
  else
    if ((clean = fopen("romanized-chart", "w")) == NULL){
      printf("Cannot open romanized-chart", "w");
      exit(1);
    }
      else {
        fprintf(clean,"\\begin{sshr}");
        fscanf(messy,"%s",unit);
        l = 0;
        while ((fscanf(messy,"%s%s%s",unit,unit2,unit3) != EOF)
          && (l <= 1994)){
          l++;
          if (strncmp(unit3,testing,2)!=0)
            fprintf(clean,"%s %s          %s\n",
                    &unit,&unit2,&unit3); }
        while (fscanf(messy,"%s%s",unit,unit2) != EOF)
          if ((strncmp(unit2,testing2,10)!=0) &&
              (strncmp(unit2,testing3,10)!=0))
            fprintf(clean,"%s             %s\n",&unit,&unit2);
        fprintf(clean,"\\end{sshr}");
```

```
        system("sshr2ks romanized-chart > korean-chart.text");
        /* korean-chart contains all the clean Korean entries now */

        system("sshr2ks specials.eng > specials.kor");
        /* specials.eng has the special cases that are not */
        /* produced by simple combinations */

        system("cat specials.kor korean-chart.text >
                korean-rewrite-rules.text");
        /* these two files are concatenated */
      }

  fclose(clean);
  fclose(messy);

}
```

Table C.3: Special.eng

```
"    "              " "
" {GgaJi}"            "GgaJi"
" {ESeo}"            "ESeo"
" {SiE}"             "SiE"
"{Je} "              "Je "
" {GeunCheoE}"       " GeunCheoE"
" {Eui}"             "Eui"
" {from}"            " "
" {Eul}"             "Eul"
" Eul"              "Eul"
" {Eun}"             "Eun"
" Eun"              "Eun"
```

Appendix D

Inflection patterns

Table D.1: Words beloging to V1 "HaDa" verbs

begin	"SiJag"
destroy	"PaGoi"
discover	"BalGyeon"
engage	"GoChag"
engaged_with	"GoChag"
fortify	"ChugSeong"
monitor	"GamCheong"
observe	"GoanChal"
pass	"TongGoa"
pay	"JiBul"
pay_attention	"JuEui"
penetrate	"ChimTu"
possess	"SoYu"
repeat	"BanBog"
report	"BoGo"
request	"YoGu"
sight	"MogGyeog"
signal	"SinHo"
think	"SaingGag"
try	"SiDo"
use	"SaYong"

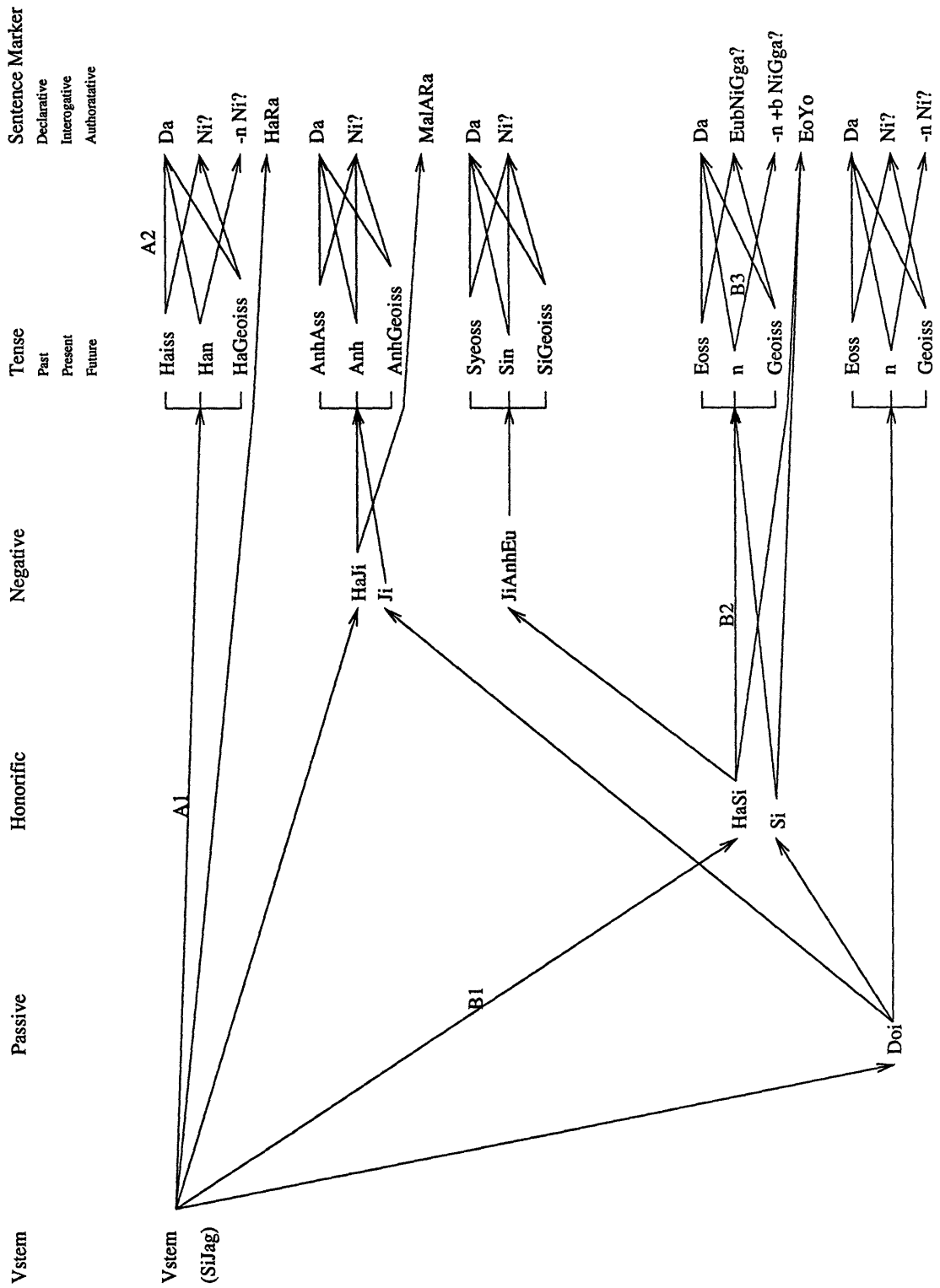

Figure D-1: Inflection patterns for V1 "HaDa" verbs

Table D.2: Words beloging to V2 verbs

bury	"Mud"

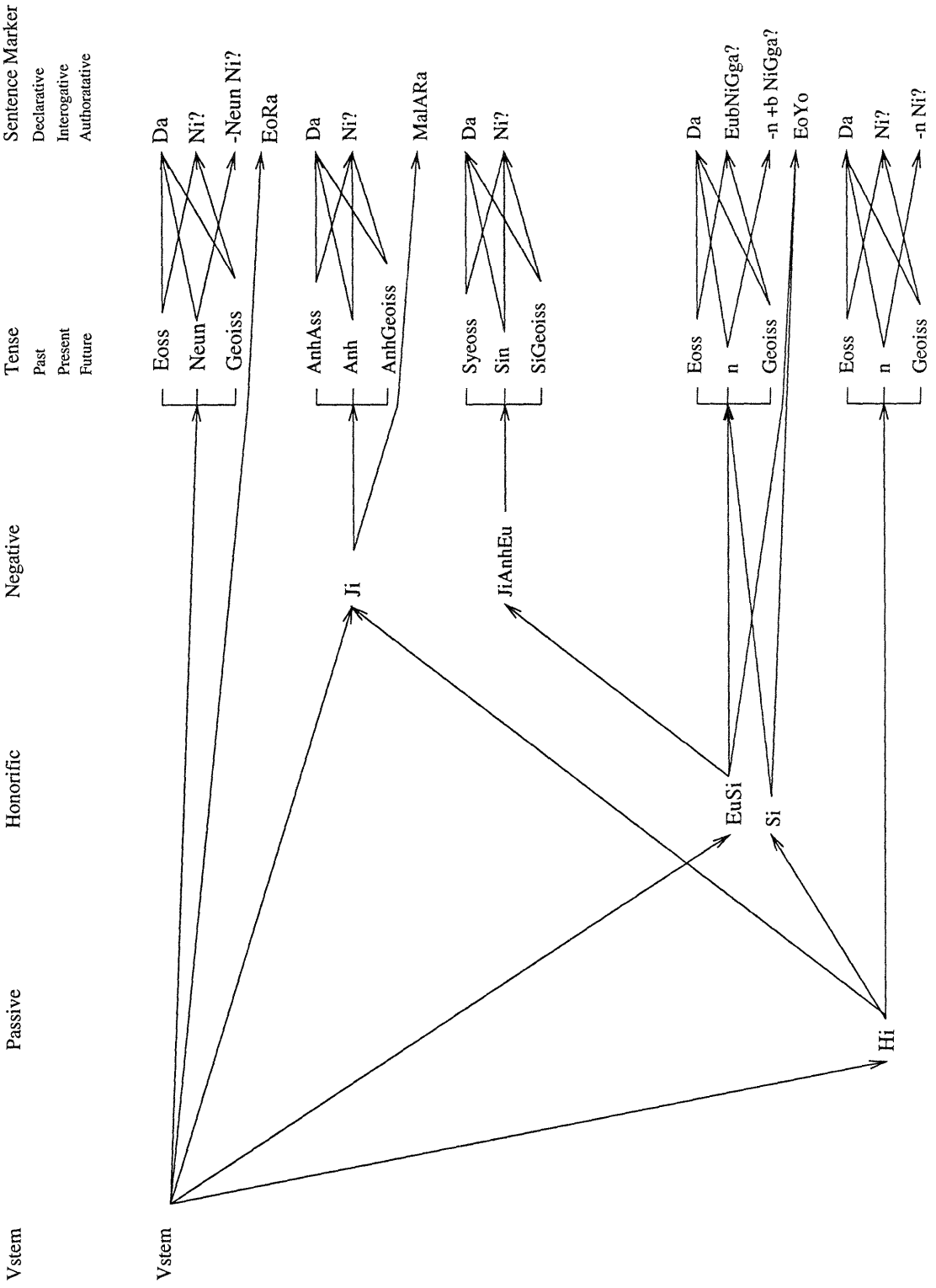

Figure D-2: Inflection patterns for V2 verbs

Table D.3: Words beloging to V3 verbs

receive	"Bad"
take_action	"Mat"
take_over	"InGyeBad"

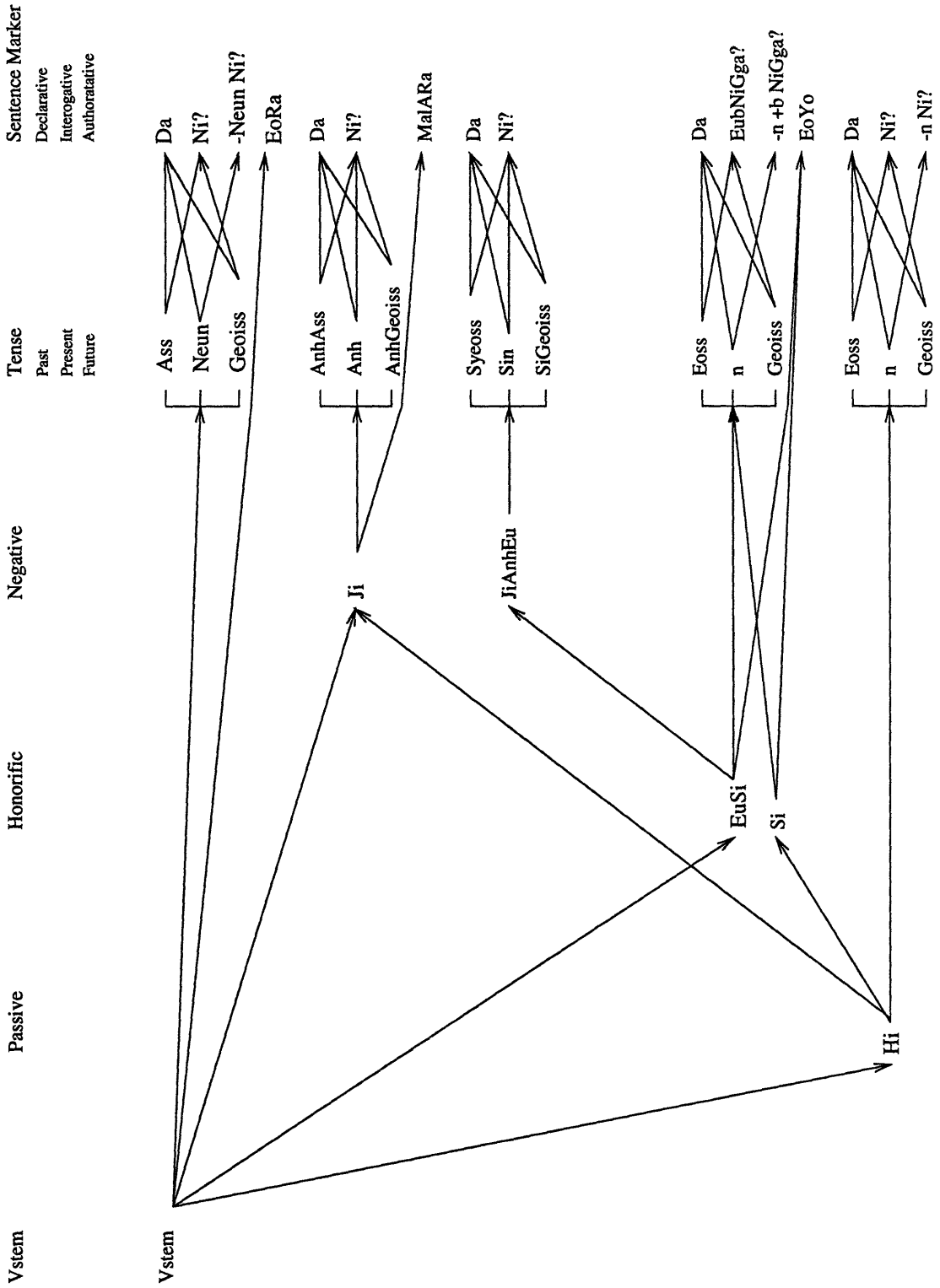

Figure D-3: Inflection patterns for V3 verbs

Table D.4: Words beloging to V4 verbs

draw	"Geus"

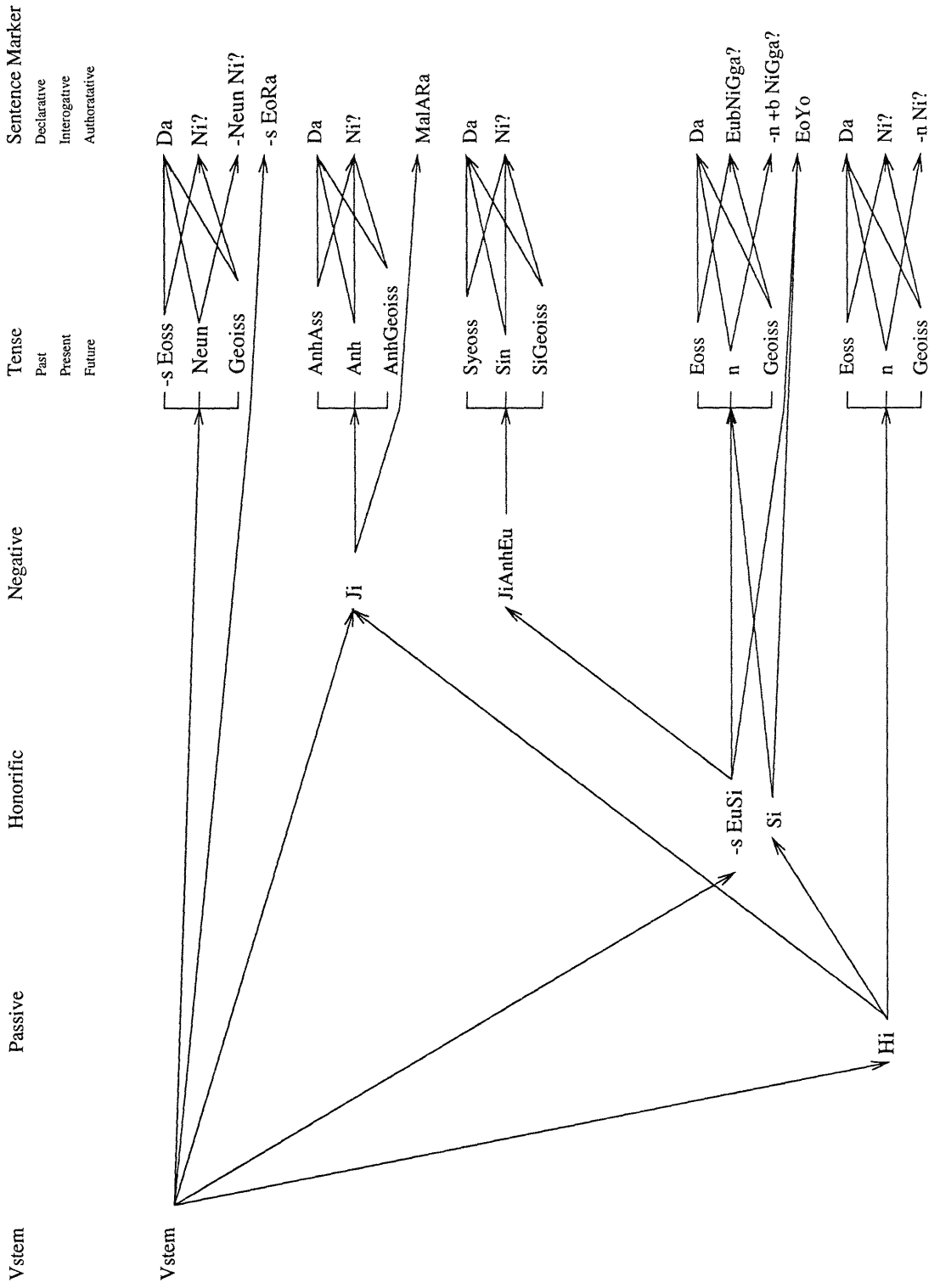

Figure D-4: Inflection patterns for V4 verbs

Table D.5: Words beloging to V5 verbs

roast	"Gub"

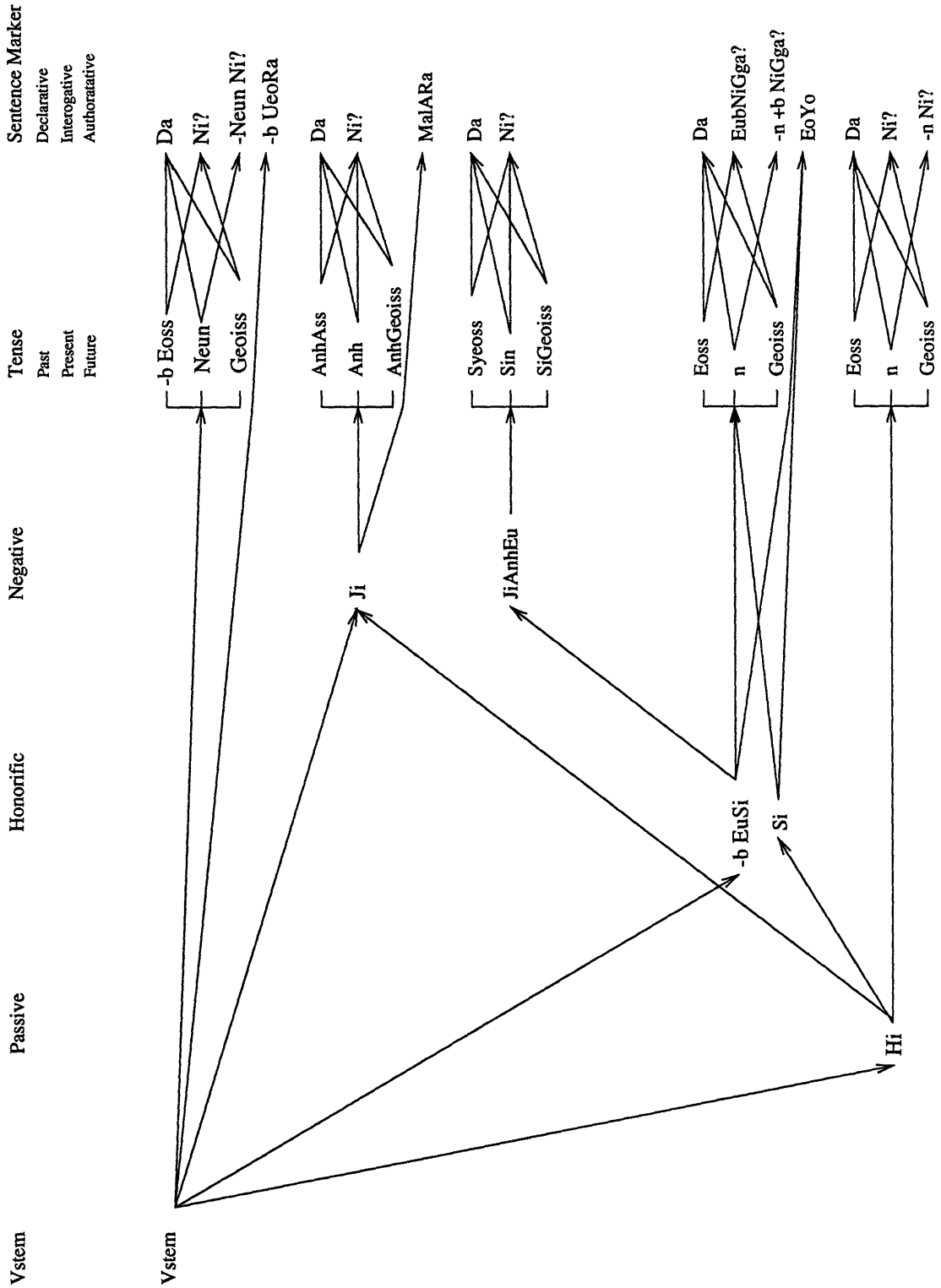

Figure D-5: Inflection patterns for V5 verbs

Bibliography

[1] S. Nirenburg, J. Carbonell, M. Tomita, and K. Goodman. *Machine Translation: A Knowledge-Based Approach.* Morgan Kaufmann Publishers, Inc., San Mateo, CA., 1992.

[2] G. Smith. *Computers and Human Language.* Oxford University Press, New York, NY., 1991.

[3] S. Ishizaki. *Generating Japanese Text from Conceptual Representation*, pages 256–279. Symbolic Computation: Natural Language Generation Systems. Springer-Verlag New York Inc., New York, NY., 1988.

[4] D. Tummala, S. Seneff, D. Paul, C. Weinstein, and D. Yang. *CCLINC: System Architecture and Concept Demonstration of Speech-to-Speech Translation for Limited-Domain Multilingual Applications.* Spoken Language Technology Workshop, Austin, TX., January 1995.

[5] S. Seneff. *TINA: A Natural Language Systems for Spoken Language Applications.* Computational Linguistics, vol. 18, no. 1, 1992.

[6] J. Glass, J. Polifroni, and S. Seneff. *Multilingual Language Generation Across Multiple Domains.* International Conference on Spoken Language Processing, Yokohama, Japan, September 1994.

[7] S. Seneff. Personal communication.

[8] G. Cho. *Highlight Senior High School Grammar.* Ji-Hag-Sa, Inc., Seoul, Korea, January 1994.

[9] H. Sohn. *Underspecification in Korean Phonology.* PhD dissertation, Univ. of Illinois, 1987.

[10] D. Tummala. Personal communication.

[11] Y. Lee. Personal communication.

[12] W. Kim and W. Rhee. *Machine Translation Evaluation.* MITRE, Seoul, Korea, January 1994.