LIESHOU : A Mandarin Conversational Task Agent for the Galaxy-II Architecture

by

Chian Chuu

B.S., Massachusetts Institute of Technology (2002)

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

Master of Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2003

Author . . . .
Department of Electrical Engineering and Computer Science
December 30, 2002

Certified by . . . .
Stephanie Seneff
Principal Research Scientist
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

LIESHOU : A Mandarin Conversational Task Agent for the Galaxy-II Architecture

by

Chian Chuu

Submitted to the Department of Electrical Engineering and Computer Science
on December 30, 2002, in partial fulfillment of the
requirements for the degree of
Master of Engineering

Abstract

Multilinguality is an important component of spoken dialogue systems, both because it makes the systems available to a wider audience and because it leads to a more flexible system dialogue strategy. This thesis concerns the development of a Chinese language capability for the ORION system, which is one of many spoken dialogue systems available within the GALAXY-II architecture. This new system, which we call LIESHOU, interacts with Mandarin-speaking users and performs off-line tasks, initiating later contact with a user at a pre-negotiated time. The development and design of LIESHOU closely followed the design of similar multilingual GALAXY-II domains, such as MUXING (Chinese JUPITER), and PHRASEBOOK (Translation Guide for Foreign Travelers). The successful deployment of LIESHOU required the design and implementation of four main components - speech recognition, natural language understanding, language generation, and speech synthesis. These four components were implemented using the SUMMIT speech recognition system, TINA Natural Language understanding system, GENESIS-II language generation system, and ENVOICE speech synthesis system respectively. The development of the necessary resources for each of these components is described in detail, and a system evaluation is given for the final implementation.

Thesis Supervisor: Stephanie Seneff
Title: Principal Research Scientist

Acknowledgments

I would like to extend my thanks and gratitude to my advisor, Stephanie Seneff, who I have had the honor of working with for the past four years. I am so proud to have been part of Orion and Lieshou since their infancy, and it has been such a great experience working and learning from her. This thesis would not have been possible without her invaluable help.

I would also like to thank everyone in the Spoken Language System group: Victor Zue, for giving me an opportunity to become a member of the SLS group; Chao Wang, who was such a great resource in helping me understand and troubleshoot the system; Min Tang, for his patience in recording one of the voices and helping with data collection for the system; Jon Yi, for helping out with the system synthesizer; Scott Cyphers, for being a great help; Michelle Spina, for helping me during the beginning of the year with generation; Jim Glass, Joe Polifroni, Marcia Davidson, and everyone else for making my experience in the SLS group memorable.

Finally, I would like to thank my loved ones for their support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Conversational systems allow humans and computers to interact via spoken dialogue, creating a more natural interface. Many current research efforts in speech technology are focused on developing *multilingual* conversational systems, i.e. conversational systems that can understand a range of languages. Multilinguality allows more users to benefit from using a system. The Spoken Language Systems group at MIT has done extensive work in developing conversational systems for English, Mandarin, and Japanese users. An evolving research effort in the SLS group utilizes the translation and synthesis capabilities of multilingual systems towards second language learning. A user would be able to improve comprehension skills through listening to simulated dialogues, or practice their speaking and listening skills through direct interaction with the system.

ORION [13] is a mixed-initiative[1] dialogue system developed for the GALAXY-II [7] architecture in the MIT Spoken Language Systems group. Users register callback tasks with ORION, and the system calls back at the appropriate time with the information that the user requested. This thesis focuses on developing a Mandarin Chinese version of ORION, including speech recognition, language understanding, language generation, and speech synthesis components. The implementation leveraged previous multilingual efforts in the SLS group, and was

[1] Goal-oriented dialogue strategy where user and system participate actively, as opposed to *system-initiative* or *user-initiative*.

greatly facilitated by existing software tools, and the capabilities of the GALAXY-II architecture to support multilinguality.

1.1 Previous Research

Conversational systems are typically applied towards information retrieval and interactive transactions, and key technologies must be implemented before the system can engage in a spoken dialogue with a user [21]. Speech recognition and language understanding components are necessary to process the input waveform, and extract a meaning from it. Generation and synthesis components are necessary to generate the spoken output. These technologies have already been developed and integrated in the GALAXY-II architecture, but further work is required to customize them to a given language and domain. In developing Mandarin ORION, we have been able to leverage previous efforts in developing Mandarin language support for other GALAXY-II domains.

1.1.1 GALAXY-II Architecture

The GALAXY-II architecture was designed with the original intention to support multilinguality, based on the assumption that it is possible to extract a language-independent semantic representation for any language. Each component involved in the process from input speech recognition to final specch synthesis has been designed to be as language independent as possible. English has been adopted as the *interlingua* for all languages.

As shown in Figure 1-1, the GALAXY recognizer, SUMMIT [27], searches a finite state transducer to transform a waveform into N-best sentence hypotheses. These are then converted to a word graph and parsed by the natural language component, TINA [12], using language-specific grammar rules. A meaning representation called a *semantic frame* is then generated. The semantic frame is then passed on to the CONTEXT RESOLUTION[2] server [5], which consults an

[2] Previously named DISCOURSE.

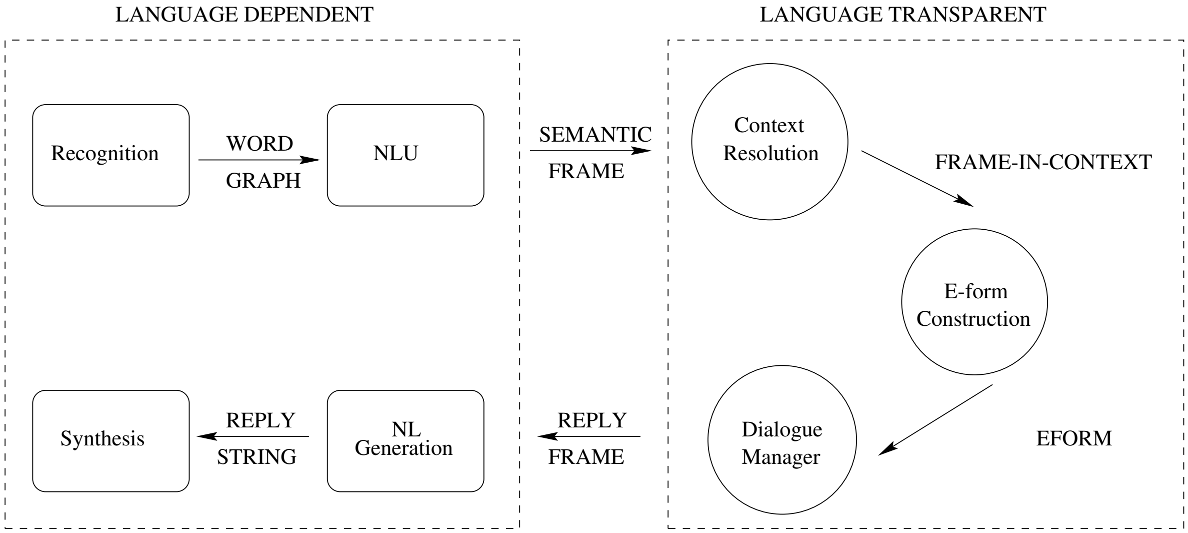


Figure 1-1: Multilingual Systems Data Flow Diagram to exhibit language transparencies

inheritance table to possibly inherit prior information from the history of the current conversation. A resulting frame-in-context is transformed into a flattened electronic form (*E*-form), encoding the meaning with a set of keys and associated values. The *E*-form provides the initial set of variables for the DIALOGUE MANAGER [10], which consults a table of rules to determine the appropriate domain routines to execute based on the *E*-form keys and values. For example, in Table 1.1, the email_message routine would be called if both the ":send_email" and ":email_address" keys were set in the *E*-form. After all the appropriate routines are executed, a final reply frame is generated. The reply frame is sent to GENESIS [2], and language-specific vocabulary and rewrite rules are applied to generate a reply string. Finally, the ENVOICE [24] speech synthesizer consults a finite state network to create a speech waveform in the target language from concatenated pre-recorded speech segments.

The components involved in the path from input waveform to final output response are designed to be as language independent as possible. The language-dependent information for the speech recognition, language understanding,

(1) :clause cphone_number & :truth_value |:phone_value	–> confirm_phone_number
(2) :clause call_me & :in_nminutes 0 & !:task_label	–> setup_immediate_call
(3) :phone_loc & !:phone	–> set_task_phonenumber
(4) :send_email & :email_address	–> email_message

Table 1.1: Examples from a dialogue control table file (shared by both LIESHOU and ORION). & stands for "AND", | stands for "OR", and ! signifies the absence of a state variable.

language generation, and speech synthesis components is encoded in external models, rules, and tables. This allows for maximal utilization of resources and provides ease in development, since a new language capability only requires specification of those models, rules, and tables. The DATABASE, CONTEXT RESOLUTION, and DIALOGUE MANAGER servers are independent of the input and output languages, i.e. they require no language specific information.

1.1.2 Mandarin GALAXY Domains

Systems in the GALAXY-II architecture which have been previously developed, providing support for Mandarin-based interaction, include YINHE (Mandarin Galaxy)[21], MUXING (weather domain)[19], and YISHU (translational phrasebook for a travel domain). Important issues that were similar to those experienced by the developers of each of these domains were the complications arising from the nature of the Chinese language, such as homophones and tone recognition (see [19, 21] for further detail).

By comparing the development of Mandarin ORION (named "LIESHOU"[3]) with that of the prior YINHE domain, developed over five years ago, it is clear how much the technologies have evolved over time. Within the last five years, an ENVOICE synthesis domain [24], a new version of generation [2], and more advanced software tools have been developed. Support for a hub scripting language for GALAXY is now available as well, thus allowing for more flexible server interaction. It was necessary to derive acoustic and language models for YINHE's

[3] Orion means "Hunter" in Greek mythology, and "Lie4 Shou3" (pinyin representation), means "Hunter" in Mandarin.

recognizer, while LIESHOU's recognizer required much less work, because the tools have become more sophisticated and models are already in place. These new tools allow us to easily create vocabulary and language models for the recognizer automatically from the NL grammar. The new version of GENESIS is much more powerful, with better control mechanisms and more intuitive knowledge specification. The ENVOICE synthesizer requires recording an extensive corpus and manual transcriptions to build a search space. Previously, an off-the-shelf ITRI[4] synthesizer required no work for the developer. Overall, with the advantage of previously developed Mandarin systems, it is possible to leverage the already existing framework, applying new tools to accelerate the development process.

1.2 Overview of ORION

ORION is being developed as a new domain of expertise for the GALAXY-II architecture, beginning about three and a half years ago as my UROP research [13]. The initial goal of my UROP project, under the supervision of Dr. Stephanie Seneff, was to create an agent that would provide a wake-up call service. Though seemingly simple, the principle of performing off-line tasks was unique, and had not yet been attempted in the SLS group. Other servers in the GALAXY system, such as the weather domain JUPITER [18] and the flight information and reservation system MERCURY [15] all assume that each task is completed as soon as the user hangs up the phone. However, ORION was designed such that the user could interact and register tasks on-line, and have the system continually maintain state information, and remember task information, long after the user has hung up the phone.

1.2.1 Orion System Configuration

ORION was made feasible as a consequence of the powerful capabilities of the GALAXY architecture. In order to retrieve information for certain tasks, ORION has to consult other domain servers; it pretends to be a standard user in requesting

[4] Industrial Technology Research Institute.

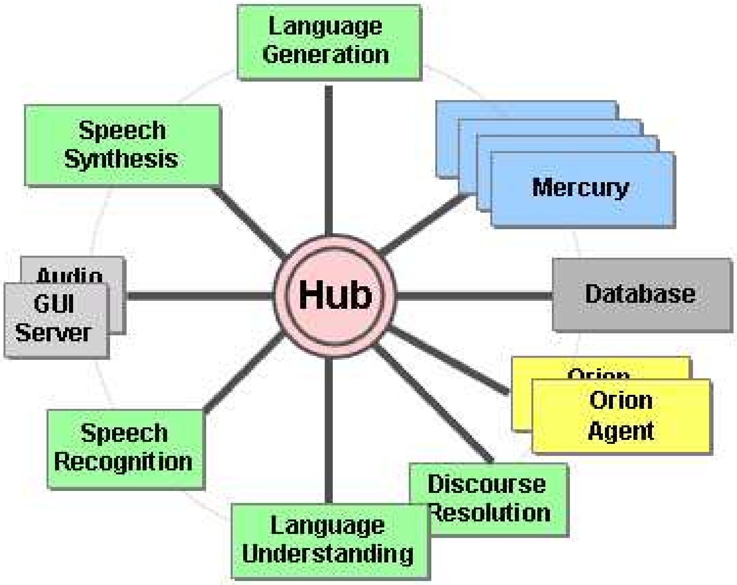


Figure 1-2: Orion System Architecture

Call me at 4 p.m. tomorrow to remind me to pick up my son at school.
Give me a wake-up call every weekday morning at 6:30 a.m. and tell me the weather for Boston.
Call me an hour before American flight 93 from New York lands in Dallas.

Table 1.2: Examples of the tasks that Orion can currently handle

this information. The ORION dialogue manager is configured as two separate servers, an "ORION USER" that deals with task registration and user interaction, and an "ORION AGENT" which monitors pending tasks and handles call backs. ORION AGENT reloads all pending user tasks at midnight, and receives alerts from ORION USER upon registration of new tasks. As mentioned before, the language transparencies of the architecture will allow LIESHOU to share the same dialogue manager as ORION.

1.2.2 Orion System Capabilities

Current tasks that ORION can handle are reminders, wake-up calls, and flight information monitoring. Due to time constraints, LIESHOU has been initially proposed to handle only two kinds of tasks, reminders and wake-up calls. There are three major phases of user interaction - user enrollment, task registration, and task execution. The user can register a single-time event, or register a task that would have repeat occurrences ("call me every day"). The system calls at the appropriate time, and is able to know that it has completed the task so as not to inadvertently perform it again on a system restart. Porting ORION to Mandarin brings up issues relating to user enrollment, since currently English users spell out the letters of their first and last name. There is no way to spell Mandarin, and characters for names are usually vocally described by referring to a commonly known phrase that contains the character, or by describing how to write the characters. This issue is talked about later in detail in Chapter 2, Speech Recognition and Language Understanding.

1.3 Approach

The next four sections describe the designing of the four language-dependent components.

1.3.1 Natural Language Understanding

It was possible to develop the input and output capabilities independently. The NLU component was implemented first, with the goal of producing the same semantic frame for a given Mandarin sentence and its English translation. Implementation required a training corpus, grammar rules, and an "actions" file mapping parse trees to meaning. An initial corpus was obtained by translating ORION's training sentences, then subsetted to include only tasks within LIESHOU's scope. LIESHOU grammar rules were inspired by analyzing the structure of the parse trees generated by ORION, and writing a set of context-free rules that would generate analogous parse trees for the Mandarin translation.

1.3.2 Generation

The generation component utilizes the GENESIS-II system. This required creating a vocabulary file, recursive ordering rules file, and a final rewrite rules file. Some entries were borrowed from YISHU, in hopes of utilizing future dialogues for a language learning tool. Words were translated from ORION's vocabulary file.

1.3.3 Recognition

Acoustic models for the recognizer were derived from the Mandarin Across Taiwan (MAT) corpus. The vocabulary and *n*-gram rules were then derived automatically from the NL grammar, and the same corpus of training sentences used to train the grammar were also used to train the recognizer statistical language model. This corpus was expanded as user data were collected.

1.3.4 Synthesis

The synthesis component required recording a carefully selected set of waveforms that would cover the system response inventory. Two native Beijing speakers, one male and one female, were chosen to be the voice talents. It is only necessary for the system to have one synthesized voice. Two voices were developed for the purpose of incorporating LIESHOU into the language learning environment, where, with two voices, a dialogue between conversational partners could be simulated.

1.4 Thesis Outline

The goal of this thesis is to port the current ORION system to Mandarin Chinese, providing a working system that will perform off-line task delegation to Chinese-speaking users. This system is called "LIESHOU." The success mark for this system will be to register and execute wake-up and reminder tasks.

The remainder of this thesis contains 5 chapters:

Implementation of the recognition and language understanding components is described in Chapter 2. An overview of the SUMMIT and TINA systems are given, and issues about the differences between the Mandarin and English language are also discussed.

The generation component is introduced in Chapter 3, and the GENESIS- II generation mechanism is described.

Chapter 4 describes the work required in implementing the synthesis component, which was configurable in two ways, either using the off-the-shelf ITRI synthesizer, or the ENVOICE [24] framework developed in the SLS group.

Evaluation methodology, data collection, and performance analyses are covered in Chapter 5. A description of the subject pool, and the collected data set is provided. An example of a user dialogue with the final working system is shown, and statistics for the task registration and callback success rates are also analyzed for the selected subjects. Analysis of each of the four implemented components is also given. Recognition and understanding error rates are calculated for the data set, the generation component is manually evaluated, and user feedback is obtained on the quality of the synthesizer.

Chapter 6 provides a summary and discussion of future work.

Chapter 2

Speech Recognition and Language Understanding

The speech recognition and language understanding components are closely coupled. The recognizer takes in a user waveform and creates a word graph to pass on to the language understanding component. The LIESHOU recognizer utilized the GALAXY SUMMIT system. In order for SUMMIT to perform recognition, acoustic models, phonological rules, an *n*-gram language model, lexicon, and syllable baseforms are required. Implementation of the LIESHOU recognizer was very easy because it was possible to leverage the domain-independent, language-dependent, resources (phonological rules, baseforms, acoustic models) from prior Mandarin systems. Existing phonological rules and baseforms were obtained from the GALAXY system MUXING, and Mandarin acoustic models had already been derived. Creating an *n*-gram language model and lexicon used to require manual effort, but due to a new capability in TINA, both could now be automatically generated from the TINA grammar. Thus the only work required in implementing a working LIESHOU recognizer was creating an NL grammar.

For language understanding, the GALAXY Natural Language (NL) component TINA, was utilized. TINA takes as input a word graph proposed by SUMMIT and applies grammar rules to extract a semantic meaning [12]. Implementing the LIESHOU language understanding components required writing grammar rules and "actions", a specification of mappings from parse tree categories to semantic frame categories.

The first section of the chapter talks about the Mandarin language, and creating a training corpus for both the recognizer and the NL component. Following, an overview of the SUMMIT recognizer is given, and a more detailed description of the LIESHOU acoustic models, lexicon, and language models is given. The second half of the chapter describes the TINA NL system, and the LIESHOU grammar rules and "actions" file.

2.1 Description of Mandarin Language

The Mandarin language consists of monosyllables formed by combinations of one of 5 lexical tones (including the less observed reduced tone) and roughly 416 base syllables. The language consists of about 1,345 unique tone-syllable pairs, which map to over 6,000 characters [21].

2.1.1 Pinyin Character Set

LIESHOU uses the standard Mandarin pinyin symbol set to represent each monosyllable. Pinyin is the system for romanizing Chinese utterances by using the Roman alphabet to spell out the sound, and the numbers [1,2,3,4,5] to represent the different tones. Special sounds are represented by certain alphabet letters. For example, the letter "x" in pinyin maps to the the "sh" sound, the letter "i" as the syllable final maps to the "ee" sound, so "xi1" would be pronounced as "shee", spoken in first tone.

2.1.2 Presence of Homophones

In Mandarin, the same tonal syllable can map to multiple unique characters, forming homophones, and it would be impossible to distinguish what character was meant without additional information. For example, the character "jia1" could either mean “home” or ”add”. This problem is solved by specifying for SUMMIT to ignore tones, and grouping pinyin characters together with underbars to form higher order units such as words and phrases. The understanding component then would apply surrounding context to determine the character. Thus word-sense disambiguation is resolved in the language understanding component.

2.1.3 Word Orderings

One difference between the Mandarin and English languages are syntactic orderings, where equivalent English and Mandarin translations will appear at different positions in the sentence. Another difference arises in the difference in frequently used words. There might not be a prevalent use for a given English word in Mandarin. For example, a typical ORION user could say, ”Call me next Friday at two thirty in the morning.” Mandarin speakers would reorder the words, leaving out the word “at”. These language differences were taken into consideration when writing the grammar rules.

Mandarin	**English**
liang3 dian3 ban4 da3 dian4 hua4 gei3 wo3 *Two clock half call telephone give me*	Call me at two thirty
you3 zi1 liao4 de5 shi2 hou4 da3 dian4 hua4 gei3 wo3 *Have information of moment call telephone give me*	Call me when you have information

Figure 2-1: Different word usages and orderings of Mandarin and English sentences with the same meaning.

2.2 Training Corpus

LIESHOU’s training corpus was obtained by manually translating domain-specific English sentences from ORION into Mandarin pinyin, keeping in mind the different Mandarin and English word orderings. A set of 390 selected sentence patterns had been used to inspire appropriate grammar rules for English ORION. These test

Mandarin	**English**
dang1 ni3 you3 zi1_liao4 de5 shi2_hou4 da3_dian4_hua4 gei3 wo3	call me when you have the information
zao3_shang4 shi2 dian3 da3_dian4_hua4 gei3 wo3	call me at 10 am
gao4_su4 wo3 dan1_fo2 de5 tian1_qi4	tell me the weather for denver
shi2 fen1_zhong1 hou4 da3_dian4_hua4 ti2_xing3 wo3 chi1_yao4	call me in ten minutes to remind me to take my medicine
wo3 de5 shou3_ji1 hao4_ma3 shi4 liu4 yi1 qi1 wu3 liu6 er4 yi1	my cell phone number is six one seven five six two one

Table 2.1: Example sentences from LIESHOU training corpus

sentences were derived from user interactions with the system during enrollment and task registration, and include different times, dates, and word ordering patterns. We focused on the more common and functional sentences and added additional sentences from Beijing speakers during the developmental process, resulting in 430 final training utterances. Some sentences, such as the names of registered system users, were entered multiple times in the training corpus to increase their likelihoods. Table 2.1 shows selected sentences from the training corpus.

2.3 SUMMIT Speech Recognition System

The SUMMIT system is a landmark-based, probabilistic, speaker-independent speech recognition system. SUMMIT utilizes both segment and boundary models to encode acoustic knowledge. In the past, it was difficult to train boundary models for a Mandarin recognizer because cross-phone boundary classes had to be manually grouped based on phonological knowledge [19]. MUXING developers were able to use a data-driven approach to automatically derive boundary classes, which LIESHOU was able to exploit.

2.3.1 Acoustic Modeling

LIESHOU's recognizer acoustic models consisted of Gaussian mixtures for diphone units derived from a large domain-independent corpus called Mandarin Across Taiwan (MAT). The MAT corpus contains close to 23,000 utterances, and has male and female speakers [23]. Chinese syllable initials and finals (i.e. onsets and rhymes) are used as acoustic model units [19]. The diphone acoustic models are mapped to phonetic surface form realizations (words) using a finite state transducer obtained after expanding the phonological rules, incorporating both the pronunciation and language model information.

2.3.2 Phonological Rules

Mandarin phonological rules were obtained from MUXING. They transform word phonemic baseforms to syllable graphs, taking into account phenomena that occur in fluent speech such as place assimilation, gemination, epenthetic silence insertion, alveolar stop flapping, and schwa reduction [9].

2.3.3 Lexicon

To define the lexicon (vocabulary) for LIESHOU, we began by translating the ORION vocabulary list, which is extensive, but domain-specific. By restricting the system to a narrow domain of expertise, we can prevent an unwieldy vocabulary size, limiting it to mostly domain-specific words. ORION has been trained from a diverse group of users for about three years, and thus the considerable vocabulary is reflective of the needs of a typical user. Additional vocabulary items were used from YISHU (the Mandarin component of the PHRASEBOOK domain), in the hope of better coverage of the reminder contents and of later utilizing LIESHOU dialogues for an YISHU-based language learning tool. The resulting vocabulary size is 643 Chinese words, covering common words used in weather queries, reminder tasks, travel, and user enrollment. Table 2.2 shows selected examples from LIESHOU's vocabulary.

da3_dian4_hua4	*call a phone number*
zhu4_ce4	*register*
ti2_xing3	*remind*
jiao4_xing3	*wake up*
zi1_liao4	*information*
bei3_jing1	*beijing*

Table 2.2: Examples from LIESHOU vocabulary

Allowable Out-of-Vocabulary Words

Out-of-vocabulary (OOV) words are allowed in LIESHOU, mainly to help in processing reminder tasks for the reminder message. A typical reminder task might be “da3 dian4 hua4 gei3 wo3 ti2 xing3 wo3 [message]” (call me and remind me [message]). This was a necessary capability for the reminder task, because the user could say anything for the reminder, which most likely will contain unknown words that would otherwise result in a failed parse. ORION handles reminder tasks by utilizing the recorded time boundaries of the hypothesized words by the recognizer, and plays back the reminder message portion of the waveform for the user at callback.

2.3.4 Baseforms

LIESHOU baseforms are idealized phonemic forms for the lexical entries. They are absent of tonal information. As mentioned before, the Mandarin language has a simple syllable structure, consisting of about 1,345 unique tone-syllable pairs. If the tone is ignored, then only 400 unique syllables are left, thus making it easier to cover the scope of the language. LIESHOU baseforms were obtained from already existing MUXING baseforms. Figure 2-2 shows examples from LIESHOU’s baseforms file.

cha2	:	ch a
chang1	:	ch ang
chang2	:	ch ang
chao1	:	ch ao
chao2	:	ch ao
che4	:	ch e
chen1	:	ch en

Figure 2-2: Baseforms from Lieshou's Syllable Lexicon

2.3.5 Language Modeling

The recognizer's language model specifies the probability of a word given its predecessors [26] , and it is typically implemented as a *class n*-gram. Previously, word classes were generated by hand, but now *n*-gram rules can be automatically generated from grammar rules using a new sentence generation capability in TINA. Training for the language model is done by parsing a corpus in TINA, which segments the utterances into words and tags them for their associated classes. The *class n*-gram model leads to better recognition performance, because it does not require as large a training set to get adaptive statistical coverage as would a *word n*-gram [17, 12].

2.4 TINA understanding system

The TINA system utilizes a top-down parse which includes an automatically trainable probability model and a trace mechanism to handle movement phenomena [17]. A Viterbi search algorithm is used to parse the word graph from SUMMIT. Context-free grammar rules and tree-to-frame mappings ("actions") are applied to produce a semantic frame [12, 8], as illustrated in Figure 2-3. The same parsing algorithm is used for all languages. A feature-passing mechanism enforces syntactic (long-distance movement and agreement) and semantic constraints [21, 12].

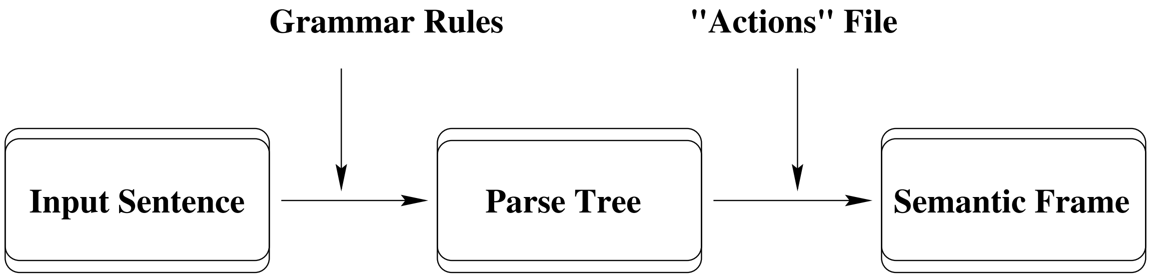


Figure 2-3: Description of data flow in TINA. An input sentence (or word graph) is parsed by grammar rules into a parse tree that encodes syntactic and semantic information. The parse tree is then processed by an “actions” file that maps the semantic categories in the parse tree to a semantic frame.

Semantic Frame Representation

The goal of the TINA parser is to generate the correct semantic frame, preserving all the necessary information from the sentence so that the system can later generate a correct response. In general, the semantic hierarchy in TINA consists of three main constituents: *clauses, topics, and predicates. Clauses* encapsulate the high level goal of the user utterances. For LIESHOU semantic frames, typical clauses could be “call_me” or “enroll”. *Topics* generally correspond to nouns (“user_name”), and usually contain one or more modifying predicates. *Predicates* are typically attributes which can be expressed syntactically as verb, prepositional, or adjective phrases [14, 12]. Predicates can be nested.

```
{c inform
   :topic {q weather
             :pred {p in
                      :topic {q city
                                :name “boston” } } } }
```

Figure 2-4: Semantic frame for the input sentence “Tell me the weather for Boston.”

2.4.1 Actions File

TINA uses an “actions” file to map parse trees to semantic frame representations, by traversing the semantic tree left-to-right and top-to-bottom, guided by syntactic roles. Each parse tree consists of parent and children branches (“nodes”) and leaves (“terminal words”). The “actions” file maps node names to semantic key names, and words that carry meaning are preserved in the semantic frame, after translation to the appropriate English equivalent in context [12].

We could take advantage of the language transparent components of the GALAXY system if we produced identical semantic frames for LIESHOU and ORION for equivalent user sentences. By allowing ORION and LIESHOU to share the same “actions” file, we could ensure that identical semantic frames would be produced if we could generate identically structured parse trees. It was found that cases such as different word orderings and other language differences sometimes differentiate the two trees. However, different word orderings did not present a problem. Thus the internal nodes were allowed to appear at different positions in the parse tree as long as we kept the hierarchical organization and the node names the same.

Keyword Mapping: English-based Interlingua

TINA provides a mechanism to map the terminals in the parse tree to translated values in the semantic frame. This mechanism takes as input a keywords mapping table, consisting of pinyin-English pairs associated with the semantic class identified with the local context in the semantic frame.

```
{c enroll
   :topic {q user_name
             :name "chu3 qian1 hui4" } }
{c enroll
   :topic {q user_name
             :name "chian chuu" } }
```

Figure 2-5: A semantic frame without keyword mapping for user login, and the corrected frame shown below after keyword mapping.

The entries for the keyword mappings file were manually generated in parallel with the grammar rules. When the parse tree and semantic frame for a given English sentence were analyzed, the node name, corresponding English semantic value, and correspondences between words from the Mandarin translation were recorded in the keyword mappings table, as illustrated in Figure 2-5[1].

2.4.2 Grammar Rules

The grammar files encapsulate both syntactic and semantic knowledge. The syntactics of the sentence are captured in the structure of the parse tree, and the semantics are encoded in the names of selected nodes of the parse tree [21, 12]. The grammar rule allows specification of optional elements (<enclosed in brackets>), exclusive alternatives (enclosed with parentheses), parent nodes (preceded with a "."), and terminal words (preceded by a "#"). Figure 2-6 shows a portion of a grammar rule associated with parsing "call me" sentences, the "call_me_phrase" node. The portion shown means that the "call_me_phrase" node should be expanded when there contains a "call" in the sentence and either the "where_phone" or "at_phone_number" nodes.

[1] Note that the name is translated into the English equivalent name that is already known to the English-based ORION system.

```
.call_me_phrase
<when_node> call_me (where_phone at_phone_number) <and_clause> <reason>
.call_me
#call <#me>
```

Figure 2-6: Portion of a grammar rule associate with parsing "call me" sentences, where optional elements are encased in brackets, exclusive alternatives are enclosed in parentheses, parent nodes are preceded by ".", and terminal words are preceded by "#".

Generating Identical Semantic Frames

The goal of the LIESHOU grammar rules was to generate an identical semantic frame from an equivalent English sentence parsed with ORION's grammar rules. ORION's grammar rules could be leveraged because extensive work had already been done to ensure correct sentence parsing. An iterative process was used. First, English sentences were run through ORION's grammar to analyze the parse tree structures. The Mandarin grammar was augmented until the Mandarin sentence could be parsed, where the nodes were rearranged, if necessary, to reflect different word orderings. If necessary, the "actions" file was modified to generate analogous semantic frames.

How to Write a Grammar Rule

An example of this rule writing process will be shown for the input sentence "call me at ten a m." The corresponding parse tree is illustrated in Figure 2-7, where the node names shown in bold are the meaning-carrying categories (specified in the "actions" file) that create the semantic frame. We can see how the parse tree maintains the syntactical order of the sentence, and also how the meaning is encoded in the node names ("me" falls under the recipient node, etc).

We want to maintain the same tree hierarchy as much as possible for the corresponding Mandarin translation "zao3 shang4 shi2 dian3 da3 dian4 hua4 gei3

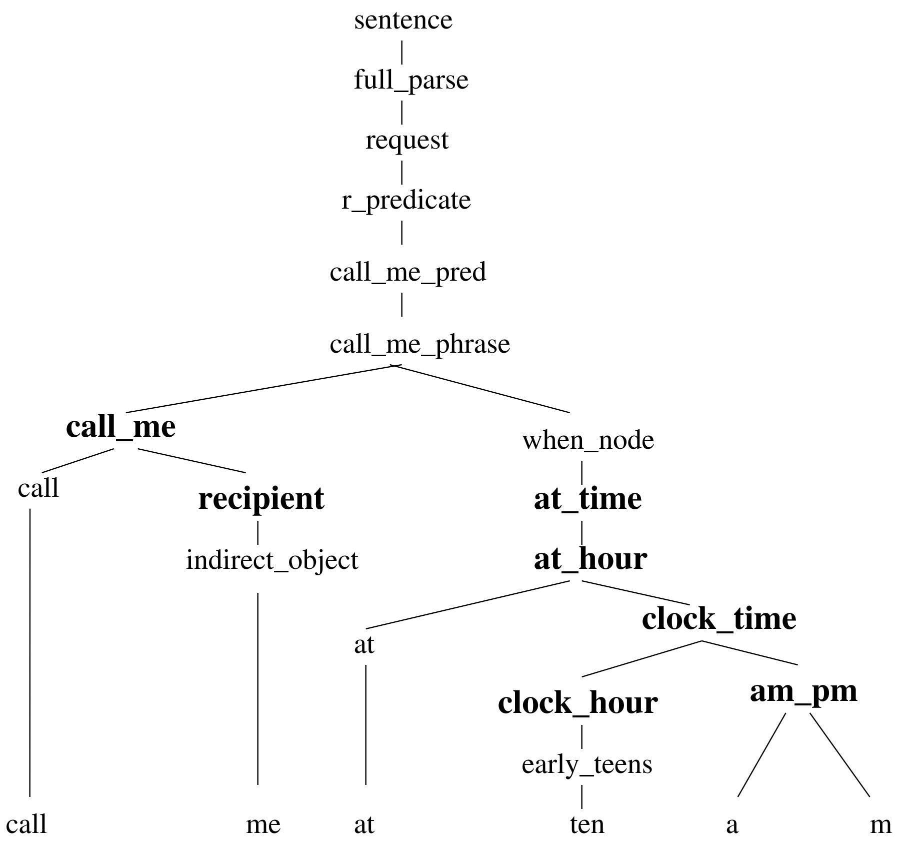


Figure 2-7: ORION Parse Tree for English sentence "Call me at ten a m"

wo3." The first step would be to try to match the Mandarin words to their English semantic equivalents, and change the terminal node mappings in the rules.

```
call -> da3 dian4 hua4
ten -> shi2
me -> wo3
a m -> zao3 shang4
```

However, this leaves us with "dian3" (a mandatory "hour" suffix) and "gei3" (give), that weren't represented in the English sentence. Semantically, the "dian3" should be associated with the node "clock_hour," so we would want to alter the clock_hour node, adding an extra terminal node associated with "dian3":

```
.clock_hour
(digits early_teens) point
.point
```

```
#dian3
```

We have to create an extra "point" node because TINA does not allow a rule to contain both terminal words and non-terminal nodes. We do not want this node in the final semantic frame, so we do not need to alter the "actions" file. For "gei3", we want it to fall under the "call_me" node, but it does not belong under the "recipient" node. Thus we create a new node called "to_recipient" to include an optional "give" node, and we alter the call_me parent node and substitute "to_recipient" in the place of "recipient."

```
.call_me
call <to_recipient>
.to_recipient
<give> recipient
.give
#gei3
```

Finally, we need to rearrange the order of the "call_me" and the "when_node" to reflect the word ordering of the Mandarin sentence under the "call_me_pred" parent node. We don't need to modify any of the other nodes, i.e. "sentence", "full_parse", etc., so we can leverage those rules from Orion. With this rule, we did not change the hierarchy of the tree, nor did we change the names associated with the "actions" file, so our new Mandarin grammar rules will generate the same semantic frame as the English grammar. Figure 2-8 depicts the final Mandarin parse tree, and Figure 2-9 shows the corresponding semantic frame identical in both grammars.

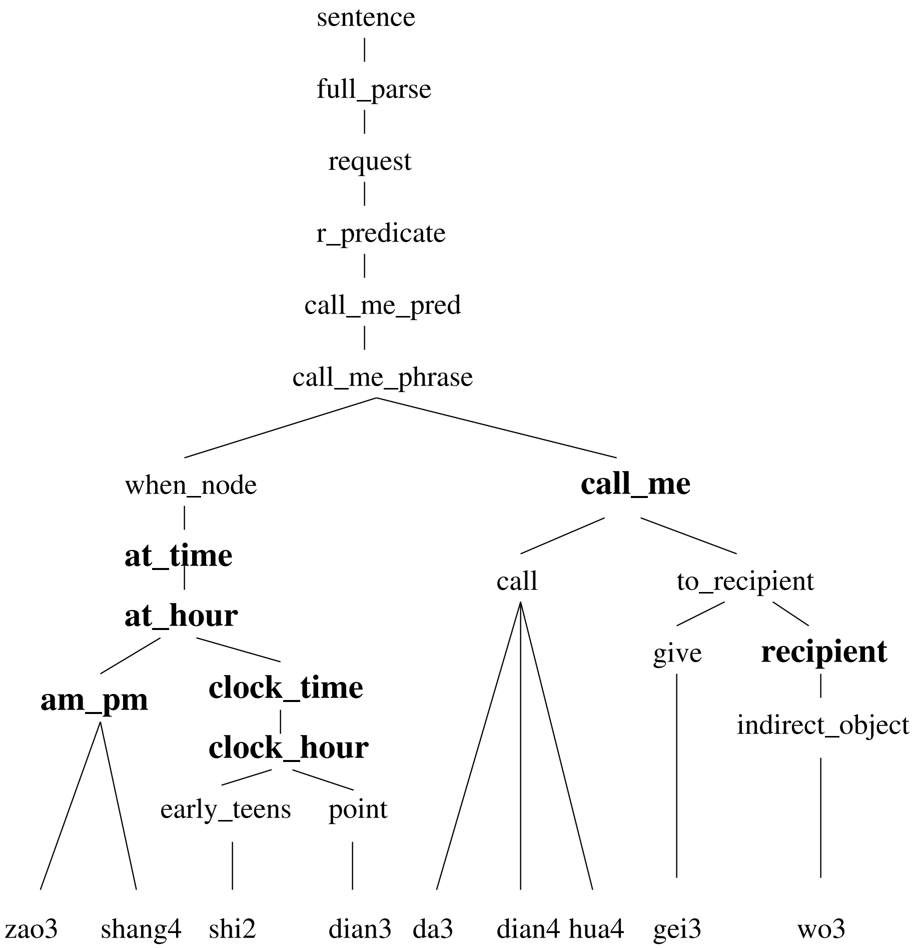


Figure 2-8: LIESHOU Parse Tree for corresponding Mandarin sentence “zao3 shang4 shi2 dian3 da3 dian4 hua4 gei3 wo3”

Mandarin Unique Sentence Patterns

Some sentences in Chinese had syntactic patterns that required writing new grammar rule patterns. We had native Beijing speakers contribute to the corpus of sentences, and obtained colloquial expressions. One type of sentence structure that is prevalent in Mandarin is illustrated in Table 2.3. In these examples, a phrase or relative clause precedes the main clause as a means of providing emphasis. We added support in the grammar to include such cases.

Additional grammar rules were written to apply to Mandarin sentences where it was possible to generate another legitimate sentence by rearranging the words in the

```
{c call_me
   :pred {p recipient
            :topic {q pronoun
                       :name "me" } }
   :pred {p at
            :topic {q time
                       :minutes 0
                       :hour 10
                       :xm "am" } } }
```

Figure 2-9: Semantic frame from parsing Mandarin “zao3 shang4 shi2 dian3 da3 dian4 hua4 gei3 wo3” sentence with LIESHOU grammar and from parsing corresponding English “call me at ten a m” with ORION grammar.

MANDARIN:	zai4 shi2 wu3 fen1 zhong1 hou4 da3 dian4 hua4 gei3 wo3 gao4 su4 wo3 tai2 bei3 be5 tian1 qi4
DIRECT TRANSLATION:	at fifteen minute after call telephone give me tell me Taipei of weather
MANDARIN:	dang1 ni3 you3 zi1 liao4 de5 shi2 hou4 da4 dian4 hua4 gei3 wo3
DIRECT TRANSLATION:	when you have information at moment call telephone give me

Table 2.3: Example of grammatical structures requiring extra topicalized phrase node (underlined portion) in Mandarin, preceding the main clause

sentence. Additional rules were also written to accommodate the many optional words that could be included in certain sentence patterns as well.

2.5 Issues in Mandarin Recognition and Understanding

Issues encountered in LIESHOU recognition and understanding were poor digit recognition, and how to identify user names during the enrollment process.

2.5.1 Poor Digit Recognition

In addition to the presence of homophones, another known issue arising in Mandarin spoken conversational systems is poor digit recognition [20]. This issue affects LIESHOU system performance, since a large portion of each session requires the recognition of phone numbers, dates, and times. For instance, “shi2” (ten) and “si4” (four) are very similar, particularly if tone is ignored. Furthermore, a sequence “yi1 yi1” (one one) is hard to distinguish from a single “yi1.” Recognition performance metrics are given in Chapter 5.

2.5.2 Mandarin User Enrollment

An issue that is unique to LIESHOU is Mandarin user enrollment. ORION requires users to enroll, to prevent the user from having to specify their contact information every time. This also enables the user to say “Call me at home”, and have the system associate that with a phone number in their profile. During ORION user enrollment, new users are asked to spell their first and last names, either vocally, or through a telephone keypad.

When a Mandarin speaker is introducing themselves to another Mandarin speaker, they usually describe their name by identifying a common word with the same character. There is no way to vocally describe character strokes to an automated system, as there is for the English-alphabet system. The current solution to this problem is to place the responsibility of name registration on the developer instead of the system. During enrollment, when the Mandarin user is speaking their first and last name, a pointer to the recorded waveform is created. After enrollment is done, I receive an e-mail notification, after which I could manually add the pinyin representation of the user’s name to the list of registered users, as well as to the NL grammar.

2.6 Summary

This chapter described the implementation of the LIESHOU speech recognition and language understanding capabilities using the SUMMIT and TINA systems, respectively. SUMMIT proposes a word graph, using as language model constraints n-gram rules automatically generated by TINA, and TINA utilizes domain-specific Mandarin grammar rules and further constraints to determine the appropriate semantic frame. The LIESHOU recognizer required minimal effort to implement, due to the ability to leverage domain-independent resources from prior Mandarin systems, as well as a mechanism to automatically generate the language model. This interaction strategy between the recognizer and understanding component allow the recognizer to suggest only acoustically promising hypotheses that are also linguistically meaningful, thus increasing the chances of a correct parse [26]. This chapter also described the resolution of homophones in the language understanding component. The performance rates of the LIESHOU recognition and understanding components are described in Chapter 5.

Chapter 3

Language Generation

The language generation system has two roles in LIESHOU: to generate an *E*-form of keys and associated values, as well as to generate a natural language paraphrase in a target language from a semantic frame. The first role, *E*-form generation, is independent of the input/output language, so we could use the existing rules from ORION for this aspect. This chapter will focus on how the GALAXY-II generation system, GENESIS-II, utilizes a lexicon, grammar rules, and a rewrite rules file to generate a Mandarin reply string. This reply string will then be passed to synthesis and spoken to the user, so it is crucial that the string be well-formed and fluent.

3.1 GENESIS-II System

The high-level architecture of the GENESIS-II system [3] consists of a kernel, written in the C language, and a linguistic catalog. The linguistic catalog is a knowledge base consisting of a lexicon, grammar, and list of rewrite rules. Each unique domain and language will tailor its own linguistic catalog to reflect the desired structure and properties of the generated reply string. For LIESHOU, outputs can appear in three distinct textual formats (pinyin, simplified and traditional Chinese characters). GENESIS-II executes recursive rewrite grammar rules on a frame with a top-down strategy, beginning with the highest level clause [2, 19] to arrive at a paraphrase.

GENESIS-II resolved many of the shortcomings of its predecessor, GENESIS [6], by providing one general framework for all types of constituents[1] allowing for greater ease in development. GENESIS-II now has more powerful mechanisms to handle movement phenomena, propagation of linguistic features, structural reorganization, and word sense specification [2].

3.1.1 Lexicon

The lexicon file maps vocabulary items (system responses, dates, known users, and other domain-specific terms that would be found inside a semantic frame) to the default generation string. Each of the three LIESHOU textual outputs requires its own unique lexicon file. For each vocabulary item, additional information is given about its part of speech (noun, etc). At times, context affects the word that is selected. GENESIS-II assures that the proper mapping is selected using a mechanism that can read context-sensitive selectors written by the developer [19, 3]. For example, Table 3.1 shows selected entries from LIESHOU's lexicon. In the case of the vocabulary entry "2", a different Mandarin word will be put in the reply string depending on the the presence of selectors (indicated by the "$SELECTOR" notation) that GENESIS-II sets from firing grammar rules depending on keys in the reply frame. The default string for "2" is "liang3". The "$:minutes" selector would have been set if "2" was meant as "two minutes past the hour", and so it would be referred to as "ling2 er4." If describing a quantity or digit in a number string, the "$:nth" selector would have been set, and thus the word sense "er4" should be selected. The "$:am" and "$:pm" selectors also indicate the different words for "2 am" (ling2_cheng2 liang) and "2 pm" (xia4_wu3 liang3) in Mandarin.

[1] clauses, predicates, topics, and keywords mentioned in Chapter 2.

Semantic Value	Pinyin Phrase
2	“liang3” $:minutes “ling2 er4” $:nth “er4” $:pm “xia4_wu3 liang3” $:am “ling2_cheng2 liang3”
5	“wu3” $:minutes “ling2 wu3” $:pm “xia4_wu3 wu3” $:am “ling2_cheng2 wu3”
no_taskphone	“wo3 ying1_gai1 da3 gei3 shen2_me5 dian4_hua4 hao4_ma3?”
something_else1	“wo3 hai2 ke3_yi3 wei4 nin2 zuo4 shen2_me5 ma5 ?”
minutes	“fen1_zhong1”
within	“hou4”

Table 3.1: Entries from LIESHOU’s lexicon that include known user names, system responses, and domain-specific terms

3.1.2 Grammar Rules

Generation grammar rules (sometimes referred to as “templates”) specify the word ordering of the constituents in the generated string. Issues that the developers of MUXING ran into was how to alter those sentences generated by GENESIS-II without completely rewriting grammar rules for those specific cases [2]. Tailoring rules to specific examples is reasonable for a small number of sentences, but the number of exceptions that slip through generation soon grows out of hand. Thus the MUXING developers found that they had to alter the parse tree itself, alter the mappings in the semantic frame, or use a newly introduced mechanism to pre-generate components from inside one or more of its descendants [19]. LIESHOU leveraged some grammar rules from MUXING and YISHU, and leveraged many domain-specific grammar rules from ORION, by reordering them as appropriate.

3.1.3 Example generation

```
REPLY FRAME:

{c orion_statement
   :missing_field “no_taskphone”
   :domain “Orion”
   :comment_tlist ( {c in_elapsed_time
                        :in_nminutes 5 } )
   :continuant {c something_else1 } }
```

Figure 3-1: Example of a reply frame and the resulting Mandarin paraphrase generated by GENESIS-II.

An example of generation is given for the example reply frame depicted in Figure 3-1. GENESIS-II begins at the top-level clause constituent, "orion_statement", and finds the corresponding grammar rule:

```
:comment_tlist :user_info :html_table :obtained_info >confirmation :missing_field
                              :missing_reminder_field
```

A string is attempted to be constructed by concatenating the constituents in the order they are listed [16]. Notation for this grammar rule includes ":" to symbolize possible keys to be accessed in the reply frame, and ">" to precede the name of another rule to be applied. Each of the keys in the grammar rule is checked for their presence in the reply frame, and in this case, ":comment_tlist" and ":missing_field" are present. The contents of the keys are recursively evaluated through the rules [16]. Here are the rules that will fire: (active rule names shown in bold)

comment_tlist	`:nth .`
in_elapsed_time	`($if :in_nhours >in_nhours >in_nminutes)`
in_nminutes	`:in_nminutes !minutes !within`
`in_nhours`	`:in_nhours !hours >and_nminutes !within`
`and_nminutes`	`:in_nminutes !minutes`

The "!" symbol indicates that the corresponding entry in the vocabulary should be evaluated. The "$if" construction says if an ":in_hours" key is present, then expand the "in_nhours" rule, otherwise expand the "in_nminutes" template. Thus the expanded reply string before substituting in vocabulary entries would consist of:

```
:in_nminutes !minutes !within :missing_field
```

GENESIS-II would then consult the lexicon for the surface form realizations for the value of “5” (wu3), “minutes” (fen1_zhong1), “within” (hou4), and “no_taskphone” (wo3 ying1_gai1 da3 gei3 shen2_me5 dian4_hua4 hao4_ma3?”). The preliminary reply string would be:

“wu3 fen1_zhong1 hou4 . wo3 ying1_gai1 da3 gei3 shen2_me5 dian4_hua4 hao4_ma3?”

The next section talks about the final step in generation, applying rewrite rules.

3.1.4 Rewrite Rules

The rewrite rules are sets of pattern matchings that act on the preliminary string from the grammar and lexicon, and process it for final refinement. These rules consist of mappings of search patterns with their replacements. Rewrite rules are written in the order they should be applied. GENESIS-II begins at the top of the file, and continues downward testing each rule until the end of the file is reached. LIESHOU's rewrite rules file is rather small, since the majority of the generation is specified programmatically in the lexicon and grammar rules. For the example given before, two rewrite rules that would apply would be “ .” to “.” (to remove the space before the period), and “_” to “ ” (since the final reply string sent to synthesis cannot contain any underbars).

Thus the final reply string for the reply frame depicted in Figure 3-1 would be:

“wu3 fen1 zhong1 hou4. wo3 ying1 gai1 da3 gei3 shen2 me5 dian4 hua4 hao4 ma3?”
(In 5 minutes. What phone number should I call?)

3.1.5 Generation for the GUI interface

There is a pinyin-to-big5 table which pairs the pinyin terms with their big5 character representations. The big5 characters were obtained using cxtermb5[2]. There are currently 1031 pinyin-to-big5 mappings in Lieshou. It was easy to distinguish the proper characters since the pinyin words were grouped together by

[2] Chinese text editor

underscores. A perl script is run to generate the unicode simplified and unicode traditional character representations of the pinyin. These unicode representations are the final output text on the visual browser window.

3.2 Summary

The generation component takes in a language independent semantic frame, and paraphrases it into a sentence in the target language, using a lexicon, grammar, and list of rewrite rules. The lexicon provides the surface mappings from the semantic tags to the appropriate target word, and the grammar rules specify the ordering of the words. The resulting string is then refined by a set of rewrite rules to arrive at the final reply string. LIESHOU's generation component was able to utilize a more powerful version of GENESIS, which resulted in greater ease in development, and well-formed Mandarin strings.

Chapter 4

Synthesis

The synthesis component takes in the reply string from the generation component, and creates a waveform to play to the user. It is essentially the “voice” of the system, and it is important that the generated waveform sound natural and understandable to the user. Prior to 1998, Mandarin GALAXY systems had utilized the ITRI synthesizer, which could generate a Mandarin waveform from any string. This required no work on the part of the developer, but the generated waveform sounded unnatural. The development of a new concatenative speech synthesis system, ENVOICE, offered the potential for far better-quality synthesis by using unit-selection to generate novel words from a sub-word corpus. LIESHOU’s synthesizer utilizes the ENVOICE system, and substantial work was required in generating the corpus, selecting two voice talents, manually transcribing the recorded sentences, and running forced alignment procedures to construct the search space. The same pronunciation baseforms that had been used for the recognizer were also used as a pronunciation graph of the lexicon. Phonological rules for the synthesizer were leveraged from YISHU. This chapter will give an overview of the GALAXY synthesizer ENVOICE, and the recording and transcription process.

4.1 ENVOICE system

The LIESHOU synthesizer utilizes the GALAXY system ENVOICE [25], which concatenates variable-length units to create natural-sounding speech. A search space is created by recording possible responses, and the recordings are manually transcribed with corresponding word sequences. The search space (also referred to as a "blob") is created by applying a pronunciation graph of the lexicon and phonological rules to the word sequences to generate phones. The phones are then composed with the search space to create *segments*. Each segment in the blob can be connected with other segments, with unique concatenation and substitution costs. The concatenation costs specify where the splices may occur, and the substitution costs specify what contexts may interchange [23]. Substitution costs would apply, for instance, if a given tonal syllable cannot be found in the blob (i.e. "ao4"), in which case another tonal syllable ("ao1", "ao2", "ao3") could be substituted at an associated cost, since it is better to have the final synthesized string be correct except for the tone than to not have the character at all.

4.1.1 Recorded Corpora

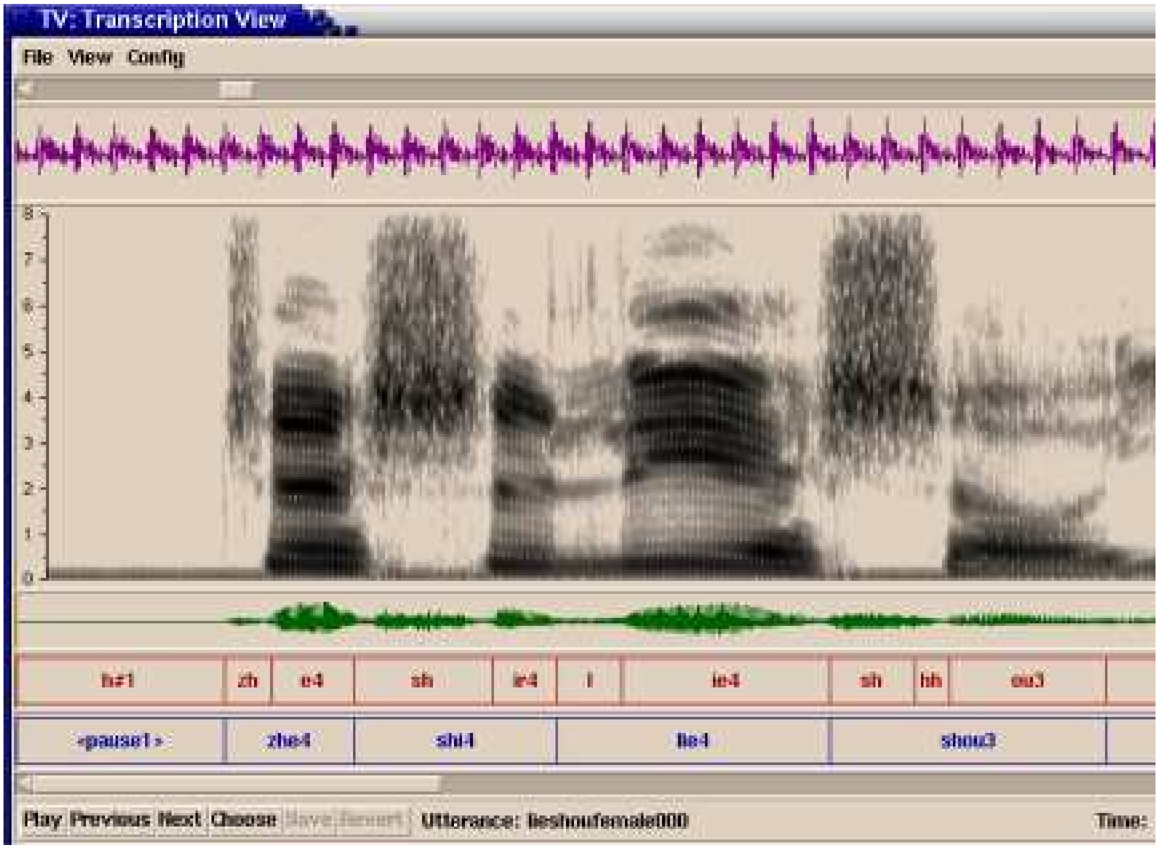
The process to create a corpus of response waveforms for LIESHOU required extensive time and effort. The initial set of system responses was obtained by looking at the vocabulary files used in the generation component, since these would be generating the reply strings that would be passed to ENVOICE. However, we also needed to cover all the possible combinations of dynamic data, such as dates, phone numbers, and times. Once the set of sentences to be recorded was obtained, we needed to find a voice talent who would be willing to volunteer their time to record the set of 238 sentences that we had selected. Ideally we would like to find one male and one female voice so that we could use the two voices to seem as conversational partners in a language learning environment currently being developed in the SLS group. We were able to find two native Beijing speakers who were willing to record their voices. By following a prompts file consisting of the 238 sentences, each sentence was read and recorded at a sample rate of 8 KHz (telephone sample rate). After each sentence was recorded, associated word (".wrd") and phone (".phn") files were created, through the use of an alignment tool that is part of the SUMMIT library. These files provide information on the temporal alignments of the words and phones, which will form concatenation units in the final blob. An example of ".wrd" and ".phn" files are depicted in Figure 4-2. The recording process was iterated during the developmental stages to accommodate the expanding vocabulary (i.e. new user names).

4.1.2 Manual Transcription Process

After the sentences were recorded, each sentence was manually transcribed into pinyin for each voice. A transcription tool allowed me to play the waveforms in a given directory one by one, and type in the pinyin representation. The transcription tool also allowed me to review the recordings. Sometimes the recording had gotten cut off, or the speaker had accidentally said a different word, so the recording had to be redone for that sentence. Transcription also had to be carefully done, since the wrong pinyin tone, or an accidental typo ("hao3" instead of "zao3") could cause the synthesizer to fail if it needed a segment that only that word could provide.

4.1.3 Forced Alignment Procedure

After the sentences had been manually transcribed, a phonetic transcription process was necessary to align the pronunciation baseforms with the words and waveforms. For each waveform, a corresponding ".phn" file was created that would have the start and end times for all the syllable initials and finals of the sentence (an example depicted in Figure 4-2). The forced alignments were automatically generated using a recognizer configured to produce forced paths [23], and the forced alignments were examined for correctness. Figure 4-1 shows a portion of the screen capture of the tool that was used to check the alignments. This tool was helpful when debugging the synthesizer through telephone conversations. For example, in one case, the generated waveform for“liang3” (“two”) seemed clipped. This required using the transcription tool to manually recalculate the start and stop times of the recorded waveform that contained the “liang3”, and repaired by manual editing of the alignments in the “.phn” file.



Utterance: lieshoufemale000

Figure 4-1: Screen capture of the forced alignment of a portion of the sentence “zhe4 shi4 lie4 shou3” (this is LIESHOU) using the transcription view tool.

The last step to creating the search space is to apply final constraints and cost functions that were already available from YISHU. Even though this process was very time-intensive, an advantage was that generating the forced paths was incremental. Every time new recordings were obtained, only forced alignments had to be performed on the new recordings, without having to run forced alignments on the entire set again. The final step in creating the blob was to apply lexical modelling using finite-state transducers, which expand the tokenized Chinese characters generated by GENESIS-II into syllable initials and finals. Additional constraints and effective cost measures, leveraged from YISHU, are also applied for better performance. The segmented corpus is then added into the FST and memory-mapped collection to generate a "blob".

Word File	**Phone File**
0 3256 <pause1>	0 3256 h#1
3256 6600 yi1	3256 6600 i1
6600 9672 yue4	6600 9672 uue4
9672 13560 er4	9672 10943 gl
	10943 13560 er4
13560 17640 shi2	13560 16200 sh
	16200 17640 ir2
17640 24120 hao4	17640 19205 h
	19205 22440 ao4
	22440 24120 gl
24120 25960 <pause2>	24120 25960 h#2

Figure 4-2: Tokenized Mandarin words and the corresponding initials and finals for a sentence recorded for LIESHOU's response inventory, **"yi1 yue4 er4 shi2 hao4"**.

4.1.4 Lexicon

A written Chinese lexicon of Tokenized Big5 characters was necessary to provide a representation that is independent of spoken dialect. For example, the pinyin representation would differ according to different dialects, etc. The lexicon consisted of mappings from LIESHOU's vocabulary (in Big5) to initial and final phones with decoupled tone. Baseforms were leveraged from the YISHU domain. The lexicon was automatically built by using a scripting tool to map entries from the Big5 representation of LIESHOU's vocabulary to their phone constituents.

4.2 Summary

The synthesizer component takes the reply string from the generation component, and creates a corresponding waveform to speak back to the user. The LIESHOU synthesizer utilized the GALAXY synthesizer ENVOICE, which has the potential to produce far better quality output than the previously used ITRI synthesizer by performing unit concatenation from pre-recorded waveforms. Because the ENVOICE system is restricted to a limited domain, and depends on domain-specific recordings to realize superior quality, much more work was required for LIESHOU's synthesizer than for previous Mandarin Galaxy systems. Significant time was required to gather sentences that would cover the response inventory, have the two selected voice talents record the sentences, manually transcribe each sentence into pinyin, run forced alignment procedures to create the search space, and check the outputs of the alignments. Leveraged work included a recognizer that could create forced paths to perform the alignments, and pronunciation baseforms from YISHU. Additional improvement on the synthesized waveforms were realized by manually editing the start and stop times of the actual waveform samples, to repair alignment errors.

Chapter 5

Evaluation

We evaluated system performance by analyzing the task success rates for four selected subjects, as well as the performance metrics for each of the four system components. This chapter will first discuss evaluation methods, and describe the data collection process. An example of a complete dialogue interaction with LIESHOU is then given, and the task success rates for the selected subjects are analyzed. Finally, performance metrics for the speech recognition (word error and sentence error rate), understanding (concept error rate), generation, and synthesis components are analyzed in detail.

5.1 Evaluation Methodology

There have been significant advancements made in the SLS group for measuring system performance. Two servers in the GALAXY-II Architecture have been specifically developed to measure system performance, BATCHMODE and EVALUATE [11]. These servers automate the evaluation process, and working together in conjunction with other GALAXY servers, can output cumulative word recognition, sentence recognition, and understanding error scores for a given data set. *Log files* make it possible for the developer to debug the system and check that the correct values are being set during the developmental process.

5.1.1 BATCHMODE Server

The batchmode server replaces the role of the user, allowing batch processing from various starting points such as waveforms or *N*-best lists of hypotheses [11]. The user utterance waveforms collected during data collection are entered in a file, and each utterance is manually transcribed using a transcription tool to generate a ".sro" file. The BATCHMODE server reprocesses those user waveforms and passes them to LIESHOU's recognizer, one by one, via the execution plan of a HUB program. The recognizer then processes the waveform and produces either an *N*-best list or a word graph. These hypotheses along with the corresponding ".sro" file, are then sent to TINA and GENESIS for parsing and paraphrasing into an E-form.

5.1.2 EVALUATE Server

The evaluation server is able to calculate error rates through comparison methods. For the recognition error rates, the standard *word error rate*, and *sentence error rate* were obtained by comparing each hypothesis with the corresponding orthographic transcription. The language understanding error rate was calculated by comparing the generated *E*-form of the transcription with the E-form that was obtained by parsing the recognizer hypotheses.

5.1.3 Log Files

Log files provide an important record of the session by preserving the activity of each session with a user. After each session, a unique time-stamped directory is created containing the user utterance waveforms, system reply waveforms, and a detailed log file. The log file contains all the dialogue turns for that given session. The information that is logged can be controlled by editing the HUB program rules, and specifying which state variables are logged when a given rule fires. The log file is the first source to consult during system debugging, since it is easy to check if the right value was set for state variables, etc.

5.2 Data Collection

Selected members of the SLS group were chosen for data collection. We chose three native-speaking users and one non-native, and divided them further into one novice, one intermediate, and two experts (shown in Table 5.1). The experts had previous experience interacting with ORION or LIESHOU, and had enrolled and registered tasks before. User 3 had been the male voice talent for LIESHOU's synthesizer. As a result, he had exposure to system dialogues, but had never directly interacted with the system, so we chose to categorize him as having intermediate domain experience. The non-experts were expected to have more spontaneous, unpredictable utterances, and the non-native was predicted to have more recognition errors due to both an accent and limited capability in the language.

	Fluency Level	**Domain Experience**
User 1	Native	Expert
User 2	Non-Native	Expert
User 3	Native	Intermediate
User 4	Native	Novice

Table 5.1: Description of fluency level and domain experience for the chosen data subjects

The non-experts were asked to enroll to create a user profile with LIESHOU. The experts had already enrolled into Orion and so the system already knew their contact information. Each of the four users were told to register at least five tasks which consisted of three weather tasks ("call me at <time> and tell me the weather for <city>") and two reminders ("call me to remind me to do <reminder message>"). These examples of tasks were given to each user, but each user was told to speak in a way that was most natural and comfortable for them. The users were told to write down how many times they successfully completed registering a task. Successful registration was defined as the moment in the dialogue when the system had verbally confirmed that the task had been registered. They were also told to record how many times the system called them back, taking into account if it was the correct time and message.

A total of 486 user waveforms were collected over one week. These 486 user utterances (from task registration and task callback interactions) were then played back and manually transcribed into pinyin using the transcription tool mentioned in Chapter 4. I also called the system during the data collection process to check if the system was healthy, and those utterances were included in evaluating recognition and understanding performance.

5.3 Task Success Rates

This section describes and gives an analysis of the task success rates for the four data subjects, and explains possible causes of task registration failure.

5.3.1 Task Registration

A task registration is considered successful if the system and user both confirmed the task details, and the system adds the task into the user's "tasks to be executed" file. The system does not write the task to the file until the contact phone number, contact time, date, and user request have been confirmed by the user. Table 5.2 shows an example of a successful task registration.

The task registration success rates are shown for each user in Table 5.5. User 4 was only able to successfully register 4 out of 13 tasks, the lowest success rate out of the four subjects. There are two related causes: the difficulties with being a novice to the system, and high recognition or understanding error rates. The registration success rate for User 4 began at 22% during the first half of the week, and improved to 67% towards the second half of the week, which was an encouraging indication of the usability of LIESHOU for novices. The relationship between task success and recognition and understanding error rates will be discussed more in detail later in the individual component evaluations.

User(U) dialogue with Lieshou(L)	Translation
L: “ni3 hao3! wo3 shi4 lie4 shou3, ni3 de5 zi4 dong4 ren4 wu4 dai4 li3. qing3 gao4 su4 wo3 ni3 de5 yong4 hu4 ming2. ru2 guo3 ni3 hai2 mei2 zhu4 ce4, qing3 shuo1, xin1 yong4 hu4.”	L: *Welcome! I am Lieshou, your automated task manager. Please tell me your user name, or if you have not yet enrolled, please say, new user.*
U: “chu3 qian4 hui1.”’	U: *chian chuu*
L: “huan1 ying2 Chu3 qian1 hui4. ni3 xiang3 an1 pai2 shen2 me5 ren4 wu4?”	L: *Welcome Chian Chuu. What task would you like to register?*
U: “qing3 zai4 wu3 fen1 zhong1 hou4 da3 dian4 hua4 gei3 wo3 gao4 su4 wo3 bei3 jing1 de5 tian1 qi4.”	U: *Please call me in five minutes and tell me the weather for Beijing*
L: “wu3 fen1 zhong1 hou4. wo3 ying1 gai1 da3 gei3 shen2 me5 dian4 hua4 hao4 ma3?”	L: *In five minutes. What number should I call?*
U: “da3 dao4 wo3 ban4 gong1 shi4.”	U: *Call me at work.*
L: “Chu3 qian1 hui4, ni3 yi3 jing1 ti2 gong4 le5 suo3 you3 xu1 yao4 de5 xin4 xi2. wo3 hui4 zai4 shi2 er4 yue4 shi2 hao4 xing1 qi1 er4 xia4 wu3 liang3 dian3 si4 shi2 liu4 fen1 da3 dian4 hua4 dao4 er4 wu3 ba1 wu3 qi1 ling2 liu4 gao4 su4 ni3 bei3 jing1 de5 tian1 qi4. dui4 ma5?”	L: *Chian Chuu, you have entered in all the necessary information. I will call you on Monday December 10 at two forty six p.m. at phone number 2585706 to tell you the weather in Beijing. Is that correct?*
U: “dui4.”	U: *Yes.*
L: “wo3 ba3 ni3 de5 ren4 wu4 yao1 qiu2 yong4 dian4 zi3 you2 jian4 chuan2 gei3 ni3 le5. hai2 you3 shen2 me5 nin2 xiang3 zhi1 dao4 de5 ma5?”	L: *I have emailed you the task information. Is there anything else I can do for you?*
U: “mei2 you3 le5 xie4 xie5.”	C: *No thank you*
L: “wu3 fen1 zhong1 hou4 zai4 liao2, zai4 jian4.”	L: *Talk to you in five minutes. Good bye.*

Table 5.2: Example of a complete LIESHOU phone dialogue consisting of a weather task registration

	Successful Registrations	**Total Attempts**	**Success Rate**
User 1	5	5	100%
User 2	5	7	71.4%
User 3	5	8	62.5%
User 4	4	13	30.8%

Table 5.3: Task Registration Success Rates for the four data subjects, calculated by dividing the total number of successful tasks by the total number of attempts

5.3.2 Task Callback

The task callback success was measured on three components: if the system tried to call, if it was at the correct time, and if the correct message was delivered. Thus, for a given registered task, a metric of "3" would correspond to the system calling at the right time and delivering the right message. The task callback success rate was only calculated with the total number of successfully registered tasks.

Callback Success Checklist	
Call Attempted	1
Correct Time	1
Correct Callback Message	1
Max success points for one task	3

Table 5.4: The breakdown of callback success to three factors: if a call was performed, if the time was correct, and if the correct task request was fulfilled.

	User Records	**Log Files**
User 1	(15/15)	(15/15)
User 2	(14/15)	(15/15)
User 3	(11/15)	(14/15)
User 4	(4/12)	(12/12)

Table 5.5: Comparison of the callback success rates according to the user records, and according to the log files for each of the four data subjects

The users had been asked to record the callback success details, and I analyzed the log files and the user records to see why the callback was not successful. Upon analysis of the log files, it was possible to find the causes of all of these “failures.” User 2 said that all the callbacks were successful, except for one where the system called at the wrong time. She had told the system to call her at 10:30, but instead the system had called her at 4:30. The system had recognized her pronunciation for “ten” (chi2) for the similar “four” (si4). The system then asked her for confirmation for “four thirty”. She mistakenly thought the system had said “ten thirty,” and thus confirmed the task details were correct.

User 3 said he never received one call, and another call had been at the wrong time. Playing back the waveforms for the callback session indicated that someone else had picked up and answered the phone. The other call had in fact been at the wrong time. Possible causes could be technical difficulties, i.e. the system was being restarted, telephony server was down, etc.

User 4 said he never received 3 calls even though he had successfully registered them. Afterwards, he realized the problem might be because he had set up his cell phone to forward his calls to another number. Upon checking the log files and listening to the waveforms for the callback sessions, it was indicated that for two of the callbacks, a forwarding service had answered (“Your call is being forwarded..”). For the other unsuccessful callback, his cell phone had been out of service, and thus another automated message was given. (“The provider you are trying to reach is unavailable..”).

Even though the system had technically successfully performed its job, to the user, the system had not been successful. We need to think of a better strategy to deal with callback issues, such as if an answering machine picks up, or if another user answers the phone. This issue is discussed further in Chapter 6, Future Work.

The next four sections will talk more in detail about performances of each of the components.

	Total Utts	WER	SER
Successful Tasks	193	15.7%	32.1%
Unsuccessful Task Registrations	108	41.4%	68.5%
Total Data Set	301	25.5%	48.4%

Table 5.6: Table of recognition error rates on subsets of 323 utterances from data collection.

5.4 Recognition Performance

Out of the 486 user utterances, only a subset was useful for evaluating recognition performance. Omitted utterances included utterances containing disfluencies[1], non-verbal utterances (silences, or pure noise), reminder messages (which were not intended to be recognized), and voicemails (where the system would call back and an answering machine or voicemail would pick up). This left a final set of 301 "clean" utterances which were run through BATCHMODE and the LIESHOU recognizer to generate the *word recognition error*[2] (WER) and *sentence recognition error rates* (SER).

5.4.1 WER and SER

The WER and SER were individually calculated for the successful tasks, unsuccessful task registrations, and then for the total data set. The *successful tasks* set consists of waveforms from successful task registration sessions and system callbacks. The waveforms from system callbacks usually were close off greetings (Thank you goodbye). The *unsuccessful task registrations* were the set of waveforms from failed registration attempts. These task registrations ended because the user prematurely hung up. The *total data set* is the sum of the *successful tasks* and the *unsuccessful task registrations*. Table 5.6 shows the WER and SER summaries for these data sets.

As shown in Table 5.6, the WER and SER from the unsuccessful task

[1] defined later in the chapter.

[2] Using the standard National Institute of Standards and Technology scoring algorithm as a library for this purpose.

registration attempts are more than double the error rates from the successful task registrations. If the system kept saying "Sorry I do not understand," this would explain why the user would choose to hang up, very possibly frustrated. Later on in this section we will talk about current work in the SLS group to analyze the "User Frustration" level.

5.4.2 WER and SER by User

Table 5.7 shows the WER and SER breakdown for each of the four data subjects.

	Fluency	Domain Experience	Total Utts	WER	SER
User 1	Native	Expert	44	5.5%	18.5%
User 2	Non-Native	Expert	44	25.1%	49.1%
User 3	Native	Intermediate	39	3.7%	12.8%
User 4	Native	Novice	94	47.8%	76.0%

Table 5.7: Table of recognition error rates on the four subjects.

The results shown in Table 5.7 are not intuitive. For example, further analysis is needed to explain why User 4 (native, novice) had higher recognition error rates than User 2 (non-native, expert), and why User 3 (native, intermediate) had lower recognition errors than User 1 (native, expert). For the first set of comparisons, it was found that User 4's speaking style and lack of experience with the system might be possible causes for his high recognition errors. He phrased his requests with long, wordy sentences, and he also used less frequently used Mandarin words. User 2 had interacted with the system and knew how to phrase her requests, so the high recognition errors were due to a limited capability with speaking the language.

It had been expected for User 1 to have the lowest recognition errors because of being a native speaker and an expert with the system. The explanation for User 3's lower WER and SER can be attributed to his lower frequency of digit use in his utterances. It is a known issue for poor recognition performance on Mandarin digits[20]. When prompted for a contact phone number or date by LIESHOU, User 1 would say a string of digits, as opposed to User 3, who would generalize it to a relative reference, such as “call me at home,” or “today.” As both of these phrasing methods are valid, our temporary solution for poor digit recognition for phone numbers is allowing the user to punch in their phone number using the telephone keypad, which is often a successful technique for getting past digit recognition errors.

5.4.3 WER and SER for Reminders

It was interesting to measure the WER and SER for the reminder messages, even though they were not intended to be recognized. Their high likelihood of containing unknown words was confirmed, as shown in Table 5.8, with nearly 100% SER (nearly every sentence was not recognized).

	Total Utts	**WER**	**SER**
Reminder Messages	16	27.9%	93.8%

Table 5.8: WER and SER statistics for 16 reminder messages from the data collected.

5.4.4 Further Recognition Performance Analysis

Recognition errors can also be attributed to other causes such as the low-quality of telephone interactions, and the frequent occurrence of disfluencies.

- **Telephone Quality Interactions:** Limited bandwidth, channel noise, and distortions can all affect recognition performance [26]. However, the telephone is the most practical and feasible displayless method of interaction for spoken dialogue systems.

- **Disfluencies :** Disfluencies include unfilled and filled pauses (“umm”, “aaah”), agrammatical sentences, and word fragments that are an unavoidable occurrence in spontaneous speech [26]. These disfluencies occurred most often when the system didn’t understand what the user is saying, and the user hesitated with their subsequent reply, possibly because they were considering how to rephrase their sentence, etc. Some native speakers had a tendency towards filler words, since it was part of their natural speaking style.

Some proposed solutions to deal with disfluencies in other conversational systems have been to introduce explicit acoustic models to deal with the filled pauses [22, 4], or to use “trash” models to deal with word fragments or unknown words [26, 1]. There is also a question of how to deal with unknown legitimate words. ORION currently handles unknown user names by having the users spell out the characters using the telephone keypad. However, this is not possible for Mandarin, and hopefully methods on how Mandarin systems can acquire new vocabulary will be discovered.

5.5 Language Understanding

To measure the performance of the NLU, the semantic frame that is generated from the original orthography (the “.sro” file), is compared with the semantic frame generated from the selected recognizer hypothesis. The metric for the differences found is described as the *concept error rate*, and the equation is shown in Figure 5-1.

5.5.1 Concept Error Rates

$$\text{Concept Error Rate} = \frac{(\#\ \text{substitutions}) + (\#\ \text{deletions}) + (\#\ \text{insertions})}{(\#\ \text{chances})}$$

Figure 5-1: Equation for Concept Error Rate.

The semantic frame from the original orthography is the “correct” frame, and the semantic frame from the selected recognizer hypothesis is the “chosen” frame. If there is a discrepancy in the semantic frames, then different types of errors could have occurred. Errors include insertions (a key-value pair was present in the hypothesis that was not in the original frame), deletions (the hypothesis was missing a concept), or substitutions (the concept in the orthography was misrecognized for another in the same class). These errors are weighted equally for the Concept Error Rate, but it can be said that insertions are the worst, since this indicates that the system is falsely hypothesizing additional information.

Table 5.9 shows the correlation between the successful task registrations and the concept error rate. The higher the concept error rate, the less likely the registration will be successful, as expected.

	Concept Error Rate
Successful Tasks	13.3%
Unsuccessful Task Registrations	61.1%
Total Data Set	27.9%

Table 5.9: Table of concept error rates for successful tasks, unsuccessful attempts, and the total data set.

Table 5.10 shows the associated system concept error rates for each of the four users.

	Concept Error Rate
User 1	4.3%
User 2	31.3%
User 3	2.8%
User 4	66.5%

Table 5.10: Table of concept error rates for each of the four data subjects.

The metrics for each of the four users all confirm a correlation between recognition error rate, understanding error rate, and task success rate, as shown in Table 5.11.

	Fluency Level	**Domain Experience**	**WER**	**SER**	**CER**	**Task Success Rate**
User 1	Native	Expert	5.5%	18.5%	4.3%	100%
User 2	Non-Native	Expert	25.1%	49.1%	31.3%	71.4%
User 3	Native	Intermediate	3.7%	12.8%	2.8%	62.5%
User 4	Native	Novice	47.8%	76.0%	66.5%	30.8%

Table 5.11: Table showing fluency level, domain experience, WER (Word Error Rate), SER (Sentence Error Rate), CER (Concept Understanding Rate), and Task Success Rate for each of the four data subjects.

5.5.2 Improving the NL grammar

It was important to isolate the valid utterances that failed to parse and add support in the grammar. It was found that, for the most part, new rules didn't have to be written, but rather small changes had to be made. For example, it was already supported in TINA to parse "bu2 yong4 le5 xie4 xie5" (no that's okay thank you), a rule that covered the sentences that a user could say at the end of a conversation. User 4 added a "le5" (modal particle) to the end of that sentence, and since that wasn't in the grammar, the sentence failed to parse. This was a simple problem to fix, and just required the addition of an optional "[le5]" to the rule.

Even though some utterances were valid Mandarin sentences, support was not provided in the grammar because the request was outside the capability of the system. For example, one user said "wo3 xiang3 gai3 ren4 wu4" (I want to change my task) during registration. We had not yet incorporated a mechanism to support editing or deleting a task via the phone. We have a capability via the GUI interface, with clickable numbers next to the user's complete list of registered tasks, so they could say "I want to edit the 5th one." As of now, LIESHOU users can use the GUI interface to edit or delete tasks, although we did not evaluate any GUI based interaction.

5.6 Response Generation

The generation component was analyzed for appropriate paraphrases of the input sentence, and appropriate reply strings based on a reply frame. The same generation evaluation method used for YINHE was used for LIESHOU. Analysis of the appropriate paraphrases was done during the developmental process, by using the correct semantic frame from TINA as the evaluation data, and the "naturalness" of the response sentence was used as the evaluation metric [21]. Evaluating that the correct reply strings are being generated was also done before the data collection process. Since the reply string can be generated through a keyword-driven approach, and the set of system responses is relatively small, it is possible to evaluate the generation component without requiring subjects. The subjects also confirmed that the system responses behaved well.

5.6.1 Improving Generation

Further work needs to be done not only to generate an *appropriate* reply string, but the most *helpful* reply string if the system runs into understanding problems. This requires either GENESIS or the DIALOGUE MANAGER (the server that oversees the generation of the reply frame) to be more advanced. It would be helpful for the system to be more active in suggestions ("Do you mean <this>?," "Perhaps it would help if you said <suggestion>"). As of now, if the system has experienced recognition or understanding errors, and has no idea of what the user has said, the response to the user is: "Sorry I do not understand what you are trying to say. Please repeat your sentence or try a shorter request." This does not help the user much, since they are not given any information about how to correct the situation. Table 5.12 shows the variety of responses obtained through data collection.

User responses to LIESHOU recognition or understanding error
LIESHOU: *"Sorry I do not understand what you are trying to say. Please repeat your sentence or try a shorter request."*
USER either: 1) Repeated their last utterance word for word. 2) Rephrased their request. 3) "I said <phrase>." 4) "I want you to <phrase>." 5) "That is not correct." 6) ”Uhhh...." 7) Hung up the phone.

Table 5.12: Examples of responses from the 4 data subjects when LIESHOU said "Sorry I do not understand what you are trying to say. Please repeat your sentence or try a shorter request".

If the system didn't understand the user's response again, then it would repeat again, "Sorry I did not understand..." This leads to a very frustrating experience for the user. The system had the highest rate of error recovery from the cases where the user paraphrased the sentences to something shorter. The system was most likely to be stuck in an error loop if the speaker was a native speaker, and repeated the exact same utterance. In a human-human interaction, it is a common reaction to repeat the original statement more slowly if the other person said "Sorry, what did you say?" For non-native speakers (User 2), it helped the system performance to repeat the original sentence more slowly, due to mispronunciation. However, native speakers who repeated the same utterance repeatedly often went into a "rejection death spiral"[26] that often resulted in a frustrated user experience.

There is current work being done in the SLS group to generate more helpful responses, and to alleviate user frustration due to recognition or understanding errors. There have been two new metrics introduced for the system performance involving recognition, understanding, dialogue management, and generation, called *User Frustration* (UF), which is the number of dialogue turns it takes for the user to repair the error, and the *Information Bit Rate* (IBR), which measure the rate that the system gains new information per turn [11]. Another research project being conducted at SLS tries to measure emotional state of the person, which might be helpful to a system so that it can be more active if it can tell that a user is getting frustrated.

5.7 Synthesis

The user experience weighs heavily on the quality of the synthesizer. It was important to measure not only that the proper waveforms were being spoken, but also that the prosodics and phonetics were acceptable. Even with perfect performance from the previous three components, if the user is not able to understand the final synthesized waveform, then the system will not have been successful.

5.7.1 Quality Rating

The four data subjects were surveyed for feedback on the synthesizer. They were asked the following three questions[3]:

1. *How would you rate the quality of the speech synthesis from 1-5?*

2. *If there were noticeable shortcomings in the quality, did you perceive sounds that are impossible to make with the human speech apparatus?*

3. *(for native speakers) If there were noticeable artifacts, did you perceive uneven pitch or intonation?*

The average quality rating was a 3. Multiple users mentioned that it was difficult to understand the synthesized phone numbers, and digits often sounded choppy and indistinguishable. This could be possibly due to alignment errors, or insufficient coverage. The pitch and intonation was noticed to be uneven at times. The rhythm of the sentence was sometimes irregular, so some words would be spoken faster than others. The native speakers also noticed that sometimes the perceived pauses were at the wrong places in the sentence.

5.8 Experts versus Non-Experts

It was interesting to analyze the performances between the experts and the non-experts, to see if there were ways we could make the system easier to use for novices. The performance success rates for the users are also analyzed for a correlation with domain expertise.

5.8.1 Experts

The performance success rates of User 1 were expected, however, the success rate of User 2 (non-native, expert) is surprising. Even though User 2 has interacted with

[3] questions composed by the ENVOICE developer, Jon Yi.

English ORION and has experience with task delegation, doesn't imply that she will be successful in registering a task. User 2 gave feedback on the data collection experience, saying that trying to register a task with LIESHOU was very helpful for her Mandarin speaking and listening comprehension skills. Task specification required her to use everyday words like dates, numbers, times, and days of the week. It was also helpful for to hear the system verbally confirm the phone number, date, and time, so that she could hear the pronunciation of the words. The performance of the LIESHOU recognizer with non-native accents is very encouraging for the future endeavor of incorporating LIESHOU into a language learning tool.

5.8.2 Non-Experts

Analyzing the interactions of the non-experts was helpful to a certain extent, since we wanted to see if there were ways to make the system easier to use for novices. Even though User 3 and User 4 were native speakers, their success rates were lower, and frustration levels were higher. The lower success rates can be attributed to recognition and understanding errors, which were likely due to their not being aware of what the system was capable and not capable of. Since they were speaking perfectly fluent Mandarin, it is natural that they would get frustrated if the system couldn't parse what they said, and they wouldn't know how to make the system understand.

User 3 was unsuccessful in one task registration because, in the middle of the conversation, the system identified him as another user, and said "Welcome <user>." He was confused, and so he hung up. Through analysis of the log files and replaying the waveforms, it was found that he had hesitated in his reply to the system, and so the user waveform consisted of pure background noise. The system had misrecognized the noise, and suggested one of the user names as a possible hypothesis. If one of the experts had been in this situation, they wouldn't have hung up, and would have known that they can just re-login to the system. We hope in future work to have more novice users try out the system to obtain more helpful feedback for improving the system usability.

5.9 Summary

The performance and error rates were typical of a system in early development, due to known issues of poor Mandarin digit recognition, limited NL coverage, and a small training corpus. As more user data are collected from diverse users with different accents and dialects, the recognition can become more speaker-independent, and the NL capabilities can be improved.

This was the first major attempt at collecting data for LIESHOU, and the training corpus was augmented considerably from all the collected user data. The system will continue to improve as more user data is collected. The task success rates also brought forth interesting issues such as the callback issues (answering machine, another person picked up), a learning curve for novices, and the relationship between recognition and understanding errors and success rates. In addition, a lot of good feedback was obtained through directly surveying the users on the performance on the system. It was also clear that LIESHOU would be useful as a second language learning tool.

Chapter 6

Summary and Future Work

6.1 Summary

Multilinguality is an important component of spoken dialogue systems, both because it makes the systems available to a wider audience and because it leads to a more flexible system dialogue strategy. This thesis describes the development of LIESHOU, an extension of the original English-based ORION system [13] to enable it to communicate with the user in Mandarin. Since its development three and a half years ago, ORION has evolved to be a reliable agent for carrying out registered tasks. From the widespread public appeal and positive feedback, there was a need to extend ORION and the functionalities that it provides to a larger audience base. Another motivation in developing LIESHOU was to utilize the task registration process for Mandarin language learning. Basic Mandarin words (dates, phone numbers, and times) are required in specifying a task, so it is a good opportunity for Mandarin beginning-level speakers to practice their speaking and listening comprehension skills, while achieving a real goal.

The design and implementation of the system was facilitated by the language independence of the GALAXY-II architecture, an architecture where the servers interact via a centralized hub [14]. The GALAXY-II architecture had been designed with the original intention of supporting multilingual development, by adopting a meaning representation called a semantic frame, and making each component as language independent as possible. Each server obtained language-dependent information by consulting external rules, files, and tables, thus some existing language resources could be leveraged from previously developed Mandarin GALAXY systems, such as MUXING [19], YINHE [21], and YISHU. LIESHOU could thus share the language-independent servers with ORION, which consisted of the CONTEXT RESOLUTION [5], DIALOGUE MANAGEMENT [10], and DATABASE servers.

This narrowed down implementation of LIESHOU to the development of four key components: speech recognition, natural language understanding, generation, and synthesis. The speech recognition and understanding components are closely coupled, and they both work together to extract the intended meaning of the input sentence. The LIESHOU system utilized the GALAXY recognizer, SUMMIT [27]. To propose a word graph for a user waveform requires Mandarin acoustic models, phonological rules, a lexicon of baseforms, and a language model. Implementation of these resources used to require extensive manual effort, but the LIESHOU recognizer leveraged already existing Mandarin acoustic models derived from the MAT corpus, and phonological rules and baseforms from MUXING. The language model was automatically generated from the NL grammar due to a new capability in the TINA NL framework [12]. The lexicon was obtained by translating the ORION lexicon, and incorporating additional vocabulary words from the YISHU translation domain. To train the recognizer and understanding components, the sentences from ORION's training corpus were translated. The recognizer achieved 25.5% WER and 48.4% SER on a data set obtained from four subjects. Many of the errors were due to high frequency of digits (it is a known issue of poor recognition on Mandarin digits), and a small training corpus.

The understanding component begins with the word graph from the recognizer, and parses the sentence into a semantic frame that encapsulates the meaning of the input sentence. The understanding component required sufficiently more work to implement, since Mandarin grammar rules had to be geared towards domain-specific sentences to generate an analogous semantic frame to one produced by ORION for the same sentence. The grammar rules for TINA combine semantic and syntactic information, and significant work was required to formulate the appropriate set of rules. This was implemented by manually parsing a typical user sentence in English through the English ORION grammar, analyzing the parse tree and corresponding English grammar rules, and augmenting the Mandarin grammar until the sentence could be parsed. The system had a 27.9% concept understanding rate for the evaluation data set.

The generation component takes in the language independent semantic frame, and either creates an E-form, or paraphrases a reply frame from the DIALOGUE MANAGER into a Mandarin reply string. LIESHOU utilizes a more powerful GALAXY generation system, the GENESIS-II system, which has better control mechanisms and consequently, better quality paraphrases [3]. Implementing the generation component required writing Mandarin grammar rules to order the information from the semantic frame into a sentence, as well as a lexicon file, which mapped semantic key values to Mandarin words. For LIESHOU, outputs can appear in three textual formats (pinyin, simplified, and traditional Chinese characters), and each one of these required a separate lexicon file. A final rewrite rules file was also needed to perform refinements to generate the final reply string, and the resulting string was evaluated for its naturalness from user feedback during system evaluation.

Finally, the synthesis component takes the reply string from the generation component, and creates a waveform to output to the user. There was considerable work in implementing the synthesizer compared to some previously developed Mandarin systems, since other systems used the off-the-shelf ITRI synthesizer to generate synthesized waveforms, in which case no work was required on the part of the system developer. However, LIESHOU utilizes the ENVOICE system [25], which requires more work, but can potentially produce far better-quality output. Implementing ENVOICE for LIESHOU required recording a set of waveforms that would cover the system response scope to create a search space that would later be used to concatenate the synthesized waveform.

The work required for the synthesizer was doubled because LIESHOU has two synthesized voices, one male and one female. Normally systems only need one synthesized voice, but we implemented two voice capabilities in LIESHOU for the purpose of incorporating LIESHOU into a language learning environment, where the two voices could seem as conversational partners. Two native Beijing speakers in the SLS group were chosen as the voice talents for LIESHOU. The synthesis feedback during system evaluation still shows that a lot of work need to be done to improve the quality, tone, and intonation of the synthesized voice.

In the final system, the task success rates, and system evaluation metrics all indicate that there is great potential for LIESHOU to be a high-performing system that can contribute to task delegation and language learning, although further improvements are required on all fronts.

6.2 Future Work

Future work includes collecting more user data to increase the robustness of the recognizer and understanding component. The LIESHOU recognizer can be configured to add word confidence scores within the N-best list, which could then be used by TINA to hypothesize when words have been misrecognized. Recognition would also benefit from the inclusion of an explicit acoustic model for the tones, although this might perform poorly for non-native speakers. We hope to improve the recognition rates on Mandarin digit recognition, as well as to make the system as speaker-independent as possible. Collecting more user data would also be beneficial in improving the NL capabilities, and the grammar would be expanded to incorporate the different ways that users can phrase their sentences.

6.2.1 Automating NL Grammars

Developing a multilingual capability will hopefully be even easier in the future. Sufficient work had to be done to write the grammar rules, by first inputting the English sentence into the English grammar, and iteratively augmenting the Mandarin grammar until an analogous semantic frame could be generated. Current work is being done in the SLS group to automate this process, and further improvements in software tools will undoubtedly reduce the effort required in future multilingual systems.

6.2.2 Resolving Callback Issues

Another issue with LIESHOU that needs to be resolved is the issue of contact failure: in case the line is busy, there is no answer, the answering machine picks up, or someone else answers the phone. Work is currently under development for Speaker ID for the ORION system, and hopefully the capability can be incorporated into LIESHOU. Another possible solution might be to have the system try calling another phone number if the callback is not received, for example, if the user specified the task phone number to be at their house, then perhaps try calling their cell phone. Another solution would be to have the system edit the task date and either call the following day saying “Yesterday I called you and I was not able to reach you. Here is the message I was supposed to deliver [ ], ” or send an email to the user with a similar message.

6.2.3 Smarter Troubleshooting

Another area for future work would be to make the system more intelligent when it encounters difficulties in understanding or recognizing the user’s utterance. Having the system say “Sorry, I do not understand. Please rephrase your request” does not give the user any help in how to better relay the information to the system. Current work in the SLS group is being done to assess the emotional state of users, so the system would be aware if the user was getting frustrated, etc. It would be interesting to incorporate that work into LIESHOU, so that the system can have different replies depending on the user’s emotional state.

6.2.4 Language Learning Environment

LIESHOU was designed not only to provide task delegational duties, but also for the hopes of incorporating user-system dialogues into a language learning tool. Since registering a task requires using everyday Mandarin words (times, dates, and phone numbers), the user-system dialogues would be useful material for language tutorials. A current project at SLS is developing a language learning environment where users can either call and directly interact with the system in practice scenarios, or listen to system-system dialogues. It is hoped to provide the two synthesized voices for LIESHOU as the two voices for the system-system dialogues. Feedback during data evaluation confirmed that it was helpful practicing the language with an interactive system, where speaking and listening skills could be improved.

Hopefully, as more data is collected, the system will improve in its capabilities, and become a truly interactive conversational system.

References

[1] A. Asadi, R. Schwartz, and K. Makhoul. Automatic modeling for adding new words to a large vocabulary continuous speech recognition system. In *Proc. ICASSP*, 1991.

[2] L. Baptist and S. Seneff. Genesis-ii: A versatile system for language generation in conversational system applications. In *Proc. 6th International Conference on Spoken Language Processing*, 2000.

[3] Lauren Baptist. Genesis-ii: A language generation module for conversational systems. Master's thesis, MIT, 2000.

[4] J. Butzberger, H. Murveit, and M. Weintraub. Spontaneous speech effect in large vocabulary speech recognition applications. In *Proc. DARPA Workshop Speech and Natural Language*, 1992.

[5] Ed Filisko. A context resolution server for the galaxy conversational systems. Master's thesis, MIT, 2002.

[6] J. Glass, J. Polifroni, and S. Seneff. Multilingual language generation across multiple domains. In *Proc. of ICSLP*, 1994.

[7] D. Goddeau, E. Brill, J. Glass, C. Pao, M. Phillips, J. Polifroni, S. Seneff, and V. Zue. Galaxy: A human-language interface to on-line travel information. In *Proc. ICSLP, 701-710*, 1994.

[8] T. Hazen, T. Burianek, J. Polifroni, and S. Seneff. Integrating recognition

confidence with language understanding and dialogue modeling. In *6th International Conference on Spoken Language Processing*, 2000.

[9] L. Hetherington. An efficient implementation of phonological rules using finite-state transducers. In *EUROSPEECH 2001*, 2001.

[10] J. Polifroni and G.Chung. Promoting portability in dialogue management. In *Intl. Conf. on Spoken Language Processing*, 2002.

[11] J. Polifroni and S. Seneff. Galaxy-ii as an architecture for spoken dialogue evaluation. In *Proc. Second International Conference on Language Resources and Evaluation*, 2000.

[12] S. Seneff. Tina: A natural language system for spoken language applications. In *Computational Linguistics*, 1992.

[13] S. Seneff, C. Chuu, and S. Cyphers. Orion: From on-line interaction to off-line delegation. In *Proc. 6th International Conference on Spoken Language Processing*, 2000.

[14] S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid, and V. Zue. Galaxy-ii: A reference architecture for conversational system development. In *Proc. ICSLP 98*, 1998.

[15] S. Seneff and J. Polifroni. Dialogue management in the mercury flight reservation system. In *Proc. ANLP-NAACL Satellite Dialogue Workshop*, 2000.

[16] S. Seneff and J. Polifroni. Formal and language generation in the mercury conversational system. In *Proc. 6th International Conference on Spoken Language Processing (Interspeech 2000)*, 2000.

[17] et al. V. Zue. From jupiter to mokusei: Multilingual conversational systems in the weather domain. In *Proc. Workshop on Multilingual Speech Communications (MSC2000)*, 2000.

[18] et al. V. Zue. Jupiter: A telephone-based conversational interface for weather information. In *IEEE Transactions on Speech and Audio Processing*, volume 8, 2000.

[19] C. Wang, S. Cyphers, X. Mou, J. Polifroni, S. Seneff, J. Yi, and V. Zue. Muxing: A telephone-access mandarin conversational system. In *Proc. 6th International Conference on Spoken Language Processing*, 2000.

[20] C. Wang and S. Seneff. A study of tones and tempo in continuous mandarin digit strings and their application in telephone quality speech recognition. In *Proc. ICSLP*, 1998.

[21] Chao Wang. Porting the galaxy system to mandarin chinese. Master's thesis, MIT, 1997.

[22] W. Ward. Modeling nonverbal sounds for speech recognition. In *Proc. DARPA Workshop Speech and Natural Language*, 1989.

[23] J. Yi. Speech synthesis: Theory and practice, 2002.

[24] J. Yi and J. Glass. Information-theoretic criteria for unit selection synthesis. In *Proc. of the 7th International Conference on Spoken Language Processing*, 2002.

[25] Jon Yi. Natural-sounding speech synthesis using variable-length units. Master's thesis, MIT, 1998.

[26] V. Zue. Conversational interfaces: Advances and challenges. In *Proc. of the IEEE*, 2000.

[27] V. Zue, J. Glass, D. Goodine, M. Phillips, and S. Seneff. The summit speech recognition system: Phonological modelling and lexical access. In *Proc. ICASSP, 49-52*, 1990.