FROM JUPITER TO MOKUSEI: MULTILINGUAL CONVERSATIONAL SYSTEMS IN THE WEATHER DOMAIN[1]

V. Zue, S. Seneff, J. Polifroni, M. Nakano, Y. Minami, T. Hazen, and J. Glass

Spoken Language Systems Group
MIT Laboratory for Computer Science
Cambridge, MA 02139 USA
and
NTT Corporation
Kyoto 619-0237, Japan

ABSTRACT

This paper describes our experiences in developing MOKUSEI, an end-to-end Japanese version of our JUPITER weather information system. JUPITER delivers weather information over the phone through a natural conversation with the user. For the most part, we were able to use the same components for recognition, understanding, and generation for Japanese that we had used for English. However, MOKUSEI motivated us to redesign our GENESIS generation system, in order to improve the quality of translations of weather reports into Japanese. MOKUSEI is currently fully functional, although it is in an early stage of development. We are in the process of collecting a large corpus of speech data from naive users in order to train and evaluate the system. This research is ongoing, and this paper represents an interim progress report.

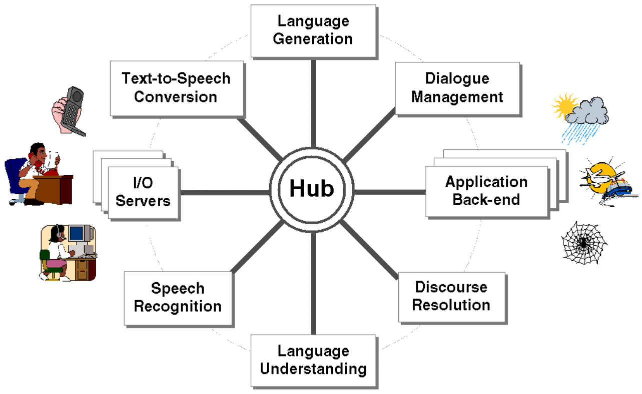

Figure 1: The Galaxy Communicator architecture.

1. Introduction

For more than a decade, our group has been conducting research leading to the development of conversational systems that enable users to access and manage information using spoken dialogue. While most of our systems have been developed for English, multilinguality has always been an important topic on our research agenda. This is partly motivated by our desire to enable information access for users who do not speak English, and our concerns to ensure that the human language technology (HLT) we develop can accommodate the diversity of different languages. In this regard, Asian languages such as Chinese and Japanese are particularly interesting because they are substantially different from English.

In 1997, we introduced the JUPITER weather information system in English [3, 12, 13]. JUPITER has been available to the public via a toll-free number in the United States since May 1997. Over the two year period since its introduction, we have collected over 400,000 utterances from over 58,000 calls, which provide a rich corpus for training and refinement of system capabilities. Since JUPITER is our most mature conversational system to date, it has become the platform for our multilingual spoken language research effort. This paper describes MOKUSEI[2], a conversational system that provides weather information in Japanese over the telephone. MOUKUSEI employs the same Galaxy Communicator architecture [6] as its English predecessor. It also utilizes most of the same HLT components, although some modifications were necessary to account for differences between English and Japanese. In addition, the weather database needed to be modified to reflect regions of greater interest to potential Japanese users. This paper describes our system development effort, paying particular attention to Japanese-specific changes to the original system. Due to space limitations, readers are referred to our other publications for a background description of JUPITER.

2. System Architecture

The overall system consists of a number of specialized servers that communicate with one another via a central programmable hub, using the Galaxy Communicator architecture [6, 9], as illustrated in Figure 1. In a telephone-based configuration, an audio server captures the user's speech and transmits the waveform to the speech recognizer. The language understanding component parses a word graph produced by the recognizer and delivers a semantic frame, encoding the meaning of the utterance, to the discourse resolution component. The output of discourse resolution is the "frame-in-context," which is transformed into a flattened E-

[1] This work was supported by NTT, and by DARPA under contract N66001-99-1-8904 monitored through Naval Command, Control and Ocean Surveillance Center. Drs. Minami and Nakano from NTT participated in this research at MIT as Visiting Researchers. The names of the authors are in reverse alphabetical order.

[2] MOKUSEI is the Japanese name for the planet Jupiter.

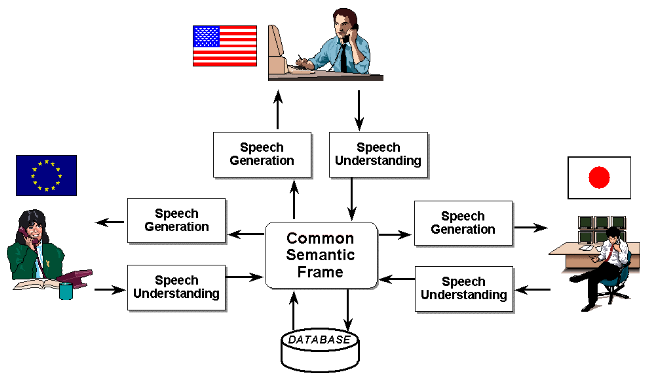

Figure 2: Schematic of multilingual system configuration: Common semantic frame captures meaning in a language transparent schema.

form (electronic form) by the generation server. This E-form is delivered to the dialogue manager, and provides the settings of the dialogue state.

The dialogue manager consults a dialogue control table to decide which operations to perform, and typically engages in a module-to-module subdialogue to retrieve tables from the database. It prepares a response frame, containing weather reports represented as semantic frames, which is sent to the generation component for translation into the target language[3]. The speech synthesizer then translates the response text into a speech waveform, which it sends to the audio server. Finally, the audio server relays the spoken response to the user over the telephone. A detailed record of the entire dialogue, including state information, is logged along with user utterances for later examination and reprocessing.

As shown in Figure 2, our approach to developing multilingual conversational systems is predicated on the assumption that it is possible to extract a *common*, language-independent semantic representation from the input, similar to the *interlingua* approach to machine translation [8]. Whether such an approach can be effective for unconstrained machine translation remains to be seen. However, we suspect that the probability of success is high for spoken language systems operating in restricted domains, since the input queries will be goal-oriented and therefore more constrained. In addition, the semantic frame may not need to capture all the nuances associated with human-human communication, since one of the participants in the conversation is a computer. Thus far, we have applied this formalism successfully across several languages and domains.

To develop a multilingual capability for our spoken language systems, we have adopted the strategy of requiring that each component in the system be as language transparent as possible. Referring back to Figure 1, the dialogue management, discourse resolution, and the application back-end are all structured so as to be independent of the input or output language. In fact, the input and output languages are completely independent from each other so that a user could speak in one language and have the system respond in another. In addition, since contextual information is stored in a language independent form, linguistic references to objects in focus can be generated based on the output language of the current query. This means that a user can carry on a dialogue in mixed languages, with the system producing the appropriate responses to each query.

2.1. Speech Recognition

Speech recognition for the MOKUSEI system is performed using the SUMMIT speech recognition system [2]. The recognizer uses a vocabulary of a little more than 1,800 *words* relevant to the weather domain. A majority of these words are names of geographic locations and words describing various weather conditions. A phonetic pronunciation for each word has been created using a standard set of Japanese phonetic units.

In order to account for phonological variations, a set of phonological rules is applied to the basic pronunciations of each word. The output is a graph of possible alternate pronunciations. For example, one set of phonological rules accounts for the deletion of /i/ and /u/, which is common in Japanese. An example of this is the word sequence *desu ka* being pronounced as /d e s k a/. The likelihoods of each alternate pronunciation are trained from training data in order to favor the more common pronunciations of each word.

For the initial version of the MOKUSEI recognizer, the acoustic models were trained entirely from English utterances. This *bootstrapping* method was required in the absence of Japanese data. These models were used in the initial version of the system and for the purpose of creating forced transcriptions of the early sets of Japanese data that were collected. As Japanese data became available, the acoustic models were retrained using a combination of English and Japanese utterances. The *hybrid* model is currently necessary because the limited amount of Japanese data that is available is not sufficient to create robust acoustic models. As more Japanese data are collected, the dependence on English data for training can be reduced, or even eliminated.

For language modeling, two types of class n-gram modeling have been examined. In the first approach, a set of 60 generic word classes were created by hand and trained using the limited amount of training sentences that are currently available. In the second approach, a class n-gram model is derived automatically from the natural language grammar used by the understanding component of the system, TINA (described next section). Both of these approaches helped alleviate sparse training data problems by sharing smoothed statistics across all words belonging to the same class.

The MOKUSEI recognizer was created in a straightforward manner using the standard tools of the SUMMIT recognizer. SUMMIT did not require any design adjustments to handle Japanese. Significant effort was required only to create the vocabulary list, pronunciations, and phonological rules, as

[3] Our English JUPITER system translates weather reports from English back into English.

(Nihon wa) doo desu ka?
((Nihon no) tenki wa)) doo desu ka?
(((Nihon no) Tokyo no) tenki wa) doo desu ka?

Figure 3: Three sentences beginning with the word "Nihon," with differing roles and parse depths for this word.

these elements could not be bootstrapped from their equivalent counterparts in the English system. The performance of the recognizer is currently hindered by a lack of training data. However, as new data become available, steady improvement in recognition accuracy can be expected.

2.2. Natural Language Understanding

Once the recognizer has proposed a word graph of promising candidates, this graph is parsed by the natural language understanding (NLU) component to produce a semantic frame. For all of our research in NLU, we have made use of the TINA system [10], which was first developed to accommodate database query domains in English. TINA utilizes a top-down parser which includes an automatically trainable probability model and a "trace" mechanism to handle movement phenomena, which are prevalent in English questions (compare "Is this restaurant on *Main Street*?" with "*What street* is this restaurant on <trace>?").

We were at first concerned that a top-down parser might not be an appropriate choice for Japanese, which is a left-recursive language. The problem is that the system must propose the entire parse column above the first word before it has seen the rest of the sentence. For example, the three sentences in Figure 3 all begin with the same word, but have significantly different parse structures, as indicated by the bracketing. A computationally expensive solution is to propose all possible parse structures and let the later evidence eliminate inappropriate ones.

We found that a much better solution to this problem was available through TINA's trace mechanism. Using this approach each new content word is parsed first in a shallow parse tree, and then later moved to a position just after the subsequent particle that defines its role. The upper columns of the parse tree are not constructed until after the appropriate role has been identified. This has the intended effect of reordering the constructs to appear right-recursive. This process is illustrated in Figure 4. The change reduced computational requirements by several orders of magnitude.

The current grammar for MOKUSEI has more than 900 categories and nearly 2,500 vocabulary entries. It was developed based on a set of sentences that were obtained from both typed and spoken inputs from native Japanese speakers as well as translations of available English queries.

2.3. Natural Language Generation

To translate weather reports into Japanese, we originally made use of our GENESIS generation system [4]. GENESIS generates text in the target language from a semantic frame, obtained by parsing the weather forecast using TINA. Generation is controlled by a message file that recursively describes the ordering of nested constituents in the semantic frame. A vocabulary file provides the mappings from the semantic tags to the appropriate target language surface realization.

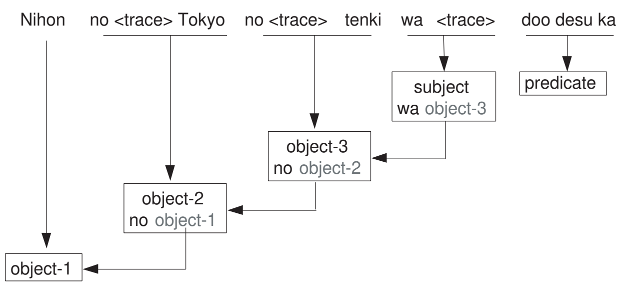

Figure 4: Illustration of use of "trace" mechanism to efficiently parse left-recursive structures in Japanese.

Mixing with or changing to rain or sleet before tapering off to patchy light rain or snow this afternoon.
Near record low temperatures with a low from upper 20s north portions to near 40 south sections near del rio.
Partly cloudy in the morning becoming mostly cloudy during the afternoon with a chance of afternoon and evening showers.

Figure 5: Examples of difficult sentences in weather reports from the U.S. National Weather Service.

Perhaps the most difficult aspect of MOKUSEI is the translation of English weather reports into Japanese. JUPITER obtains its weather information from four different Web sites, plus a continuous stream of real-time information from a satellite feed. A large portion of the weather for the United States is obtained from the National Weather Service. These reports, which are prepared manually by weather forecasters, can be quite rich and structurally complex. They thus provided a challenging task of fluent translation into Japanese. Some example sentences are given in Figure 5.

In attempting to generate well-formed Japanese from the semantic frames for the weather reports, we encountered several situations where GENESIS was unable to produce a natural Japanese translation. In some cases, we were able to solve the problem by changing the parsing grammar, for example by distinguishing semantically between "in_time" and "in_location." However, we found that more explicit control was needed over the choice of both the correct alternative translation of a given word and the correct ordering of certain constituents that were too deeply embedded.

To solve these problems, we have recently completely redesigned our GENESIS system, thus providing far greater flexibility in translation tasks. Additional features of this GENESIS-II system [1] include better control over the ordering of constituents (e.g., noun phrase modifiers), and the capability of selecting for context-dependent lexical realization. These additional features were essential for high quality gen-

low near -5	→	saitee kion mainasu 5 do *kurai*
near record high	→	*hotondo* kirokuteki na saikoo kion
near 100 percent chance of rain	→	ame no kakuritsu *yaku* 100 paasento

Figure 6: Three different usages for the word "near" in the weather domain, with their corresponding translations into Japanese.

rain (possible (in the morning))	[English original]
((morning in) possible) rain	[Japanese ordering]
((gozenchuu ni) kanoo na) ame	[*nonsense*]
((gozenchuu ame no) kanoosee) ga arimasu	[*fluent*]
((morning rain of) possibility) SBJ exist	[English equivalent]

Figure 7: An example weather phrase whose literal translation is nonsensical, along with the appropriate translation that is produced by our system.

eration in Japanese.

One powerful new feature is the ability to select different word senses of a vocabulary item based on semantic context. For example, the lexical entry for the word, "near" is as follows:

near "kurai" $:record "hotondo" $:percentage "yaku"

Examples of these three distinct usages of "near" are shown in Figure 6.

Often, a literal translation, even properly reordered, is inappropriate, as illustrated in Figure 7, with the semantic frame for the original sentence shown in Figure 8. Notice that fluent translation requires a major reorganization of the sentence structure. In particular, the word "rain" ("ame") needs to appear *between* the IN_TIME predicate ("gozenchuu") and the word "possibility" ("kanoosee"), which is impossible if each frame is required to generate its entire string as a single unit.

GENESIS-II provides a powerful "pull" mechanism that over-

```
{c weather_event
   topic:
     {q precip_act name: "rain"
        pred: {p possibility
                 qualifier: "possible"
                  pred: {p in_time
                           topic:
                              {q time_of_day
                                 name: "morning"
                                 quantifier: "def"
                              }
                         }
                 }
       }
}
```

Figure 8: Semantic frame for the sentence, "rain possible in the morning," showing the embedded IN_TIME predicate.

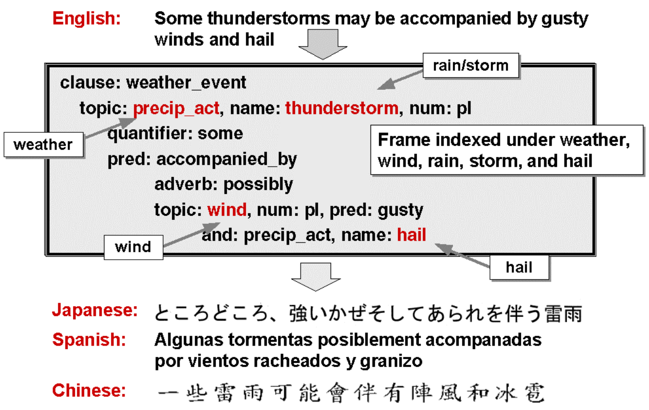

Figure 9: Example semantic frame for the sentence, "Some thunderstorms may be accompanied by gusty winds and hail," illustrating content processing in MOKUSEI, as well as our Spanish and Chinese systems.

comes this problem. Any constituent can pregenerate by name any other constituent among its grandchildren (< −−), or even among its however deeply nested descendants (<==). In our example, the weaker pull (< −−) is sufficient, and is accomplished through the following rule:

precip_act   < −−in_time ... :name ... probability

resulting in the fluent translation shown in Figure 7.

For MOKUSEI, we have created nearly 500 generation rules, along with a generation vocabulary of about 3000 entries. Most of the time, the weather reports are produced in fluent Japanese, or, at least, can be properly understood.

2.4. Dialogue Management and Synthesis

MOKUSEI makes use of a dialogue manager that is identical to the one used for JUPITER. The only change we introduced was to have it default to the Celsius scale for temperatures when the language is Japanese. This requires a mechanism to interpret frequent references to ranges of temperature, such as "high upper 60s". Aside from this one feature, the dialogue manager does not need to know either the input or output languages. It prepares a response as a semantic frame, which later gets translated into text in the target language by the generation system.

For Japanese synthesis, we made use of FLUET, a software synthesizer provided by NTT Cyber Space Laboratories [7]. We compiled it with a wrapper that allowed it to talk to the hub and communicate with the other servers via the standardized protocol.

2.5. Content Processing

JUPITER updates its weather database three times a day from the four Web sites, and updates information about current temperature, humidity, and pressure continuously from its satellite feed. The update involves several steps. After the new weather reports are retrieved from the Web sites, they are all parsed into semantic frames using our TINA NLU system. The frames are then examined automatically for semantic content, and indexed under all weather categories that are applicable. The relational database tables are then updated to reflect the new weather reports. An example of content processing in MOKUSEI is given in Figure 9.

The database for MOKUSEI is nearly identical to the one used by JUPITER. However, since JUPITER knew of only six cities in Japan (Kyoto, Nagano, Osaka, Sapporo, Tokyo, and Yokohama), we expanded the number of Japanese cities to 46 for MOKUSEI. We also provided in a geography table information about the prefecture for each city. The weather information for the expanded Japanese set is obtained from a Web site.

2.6. Data Collection

Data collection is a vital part of conversational system development. Once a preliminary version of every component is available, users can be invited to talk to the system to obtain information. The system records all of the users' speech as well as a detailed log file representing the interaction. Later perusal of the log file reveals problems that can then be repaired by system developers. The users' queries are transcribed, and the text is used to guide expansion of the grammar rules for the NLU component, as well as the language model for recognition. The waveforms can be used to develop improved versions of the acoustic models.

One of the most successful aspects of our JUPITER system has been the amount of data we have collected over the course of the three years it has been deployed. The weather domain is of universal interest, and, with continual updates on dynamic content, users will call multiple times a week or even within the course of a single day. Relying exclusively on word-of-mouth for recruitment of subjects, we have collected over 400,000 utterances in JUPITER's short lifetime, from nearly 50,000 callers. Based on these numbers, we wanted to deploy an initial version of MOKUSEI to begin collecting real data as quickly as possible. The software and hardware for data collection currently reside at MIT, while the callers are recruited in, and call from, Japan.

Before we began data collection, we first modified MOKUSEI to know about a different set of cities than JUPITER (see Section 2.5). We also made changes to the scripts that harvest and process the forecast data, based on the mismatch in time zones between where the forecast information was obtained (i.e., the U.S.) and where users were calling from (i.e., Japan). Relative terms such as "today" and "tomorrow" become quite problematic with a 13- to 14-hour time difference. The weather data we harvested is inconsistent in its own use of these terms (e.g., updates seem to happen hapharzardly and are sometimes synchronized to the local time in Japan and sometimes synchronized to a time zone in the U.S.) and we found that the most reliable solution was to make the weather update for Asian cities occur at around 6:00 p.m. U.S. Eastern time. We feel that ultimately the solution will be to determine the time zone that the subject is calling from and use that to make decisions on how to use relative time expressions.

To date, we have collected over 130 calls from naive users of the system. The total number of utterances from these calls is approximately 1,700. The higher number of utterances per call, more than double that of JUPITER [3], is presumably due to the fact that speakers were encouraged to speak as many utterances as possible. These data, together with some 2,000 *read* utterances that we have collected and transcribed, are being used for system training, development, and debugging. Following a large scale data collection effort planned for this fall at NTT in Japan, we will be able to conduct formal evaluation, much the same way that we have done for JUPITER [12].

3. Summary

This paper describes MOKUSEI, a weather information system for Japanese speakers that enables them to obtain real weather information for cities worldwide by conversing with the system in Japanese. MOKUSEI is completely functional, but all of its components are in a preliminary stage of development. As we refine the system to accommodate the new data that we are collecting from users, it will continue to improve in recognition and understanding performance. We have expanded the capabilities of our GENESIS generation system such that it should now be possible to generate high quality translations of the English weather reports. We are in the process of refining the generation rules for MOKUSEI to make further use of the new features of GENESIS.

We found this research to be an excellent mechanism to enable us to expand the capabilities of our conversational system components towards the ambitious ultimate goal of universal language capability.

4. Acknowledgements

Lauren Baptist has contributed significantly to the development of GENESIS-II, which is essential for high quality response generation in MOKUSEI. We would alos like to acknowledge the many colleagues from NTT who have contributed to our data collection effort.

5. REFERENCES

1. L. Baptist and S. Seneff, "GENESIS-II: A Versatile System for Language Generation in Conversational System Applications," *Proc. ICSLP*, Beijing, China, Oct. 2000.

2. J. Glass, J. Chang, and M. McCandless, "A Probabilistic Framework for Feature-based Speech Recognition," *Proc. ICSLP*, 2277–2280, Philadelphia, PA, 1996.

3. J. Glass, T. Hazen, and I. Lee Hetherington, "Real-time Telephone-based Speech Recognition in the JUPITER Domain," *Proc., ICASSP*, 61–64, Phoenix, AZ, 1999.

4. J. Glass, J. Polifroni and S. Seneff, "Multilingual Language Generation Across Multiple Domains," *Proc. ICSLP*, 983-986, Yokohama, Japan, 1994.

5. J. Glass, G. Flammia, D. Goodine, M. Phillips, J. Polifroni, S. Sakai, S. Seneff, and V. Zue, "Multilingual Spoken-language Understanding in the MIT VOYAGER System," *Speech Communication*, 17(1-2), 1–18, 1995.

6. 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," *Proc. ICSLP*, 707-710, Yokohama, Japan, 1994.

7. K. Hakoda, T. Hirokawa, H. Tsukada, Y. Yoshida and H. Mizuno, "Japanese Text-to-Speech Software based on Waveform Concatenation Method," AVIOS '95, pp.65-72, 1995.

8. W. Hutchins and H. Somers, "An Introduction to Machine Translation," Academic Press, 1992.

9. S. Seneff, E. Hurley, R. Lau, C. Pao. P. Schmid, and V. Zue, "Galaxy-II: A Reference Architecture for Conversational System Development," *Proc. ICSLP*, 1153–1156, Sydney, Australia, 1998.

10. S. Seneff, "TINA: A Natural Language System for Spoken Language Applications," *Computational Linguistics*, 18(1), 61–86. 1992.

11. C. Wang, J. Glass, H. Meng, J. Polifroni, S. Seneff, and V. Zue, "YINHE: A Mandarin Chinese Version of the GALAXY System," *Proc. Eurospeech*, 351–354, Rhodes, Greece, 1997.

12. V. Zue, S. Seneff, J. Glass, J. Polifroni, C. Pao, T. Hazen, and L. Hetherington, "JUPITER: A Telephone-Based Conversational Interface for Weather Information," *IEEE Trans. Speech and Audio Proc.*, 8(1), 85–96, 2000.

13. V. Zue, S. Seneff, J. Glass, L. Hetherington, E. Hurley, H. Meng, C. Pao, J. Polifroni, R. Schloming, and P. Schmid, "From Interface to Content: Translingual Access and Delivery of On-Line Information," *Proc. Eurospeech*, 2227–2230, Rhodes, Greece, 1997.