A Bilingual VOYAGER System[1]

J. Glass, D. Goodine, M. Phillips, S. Sakai[2], S. Seneff, and V. Zue[3]

Spoken Language Systems Group
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

ABSTRACT

This paper describes our initial efforts at porting the VOYAGER spoken language system to Japanese. In the process we have reorganized the structure of the system so that language dependent information is separated from the core engine as much as possible. For example, this information is encoded in tabular or rule-based form for the natural language understanding and generation components. The internal system manager, discourse and dialogue component, and database are all maintained in language transparent form. Once the generation component was ported, data were collected from 40 native speakers of Japanese using a wizard collection paradigm. A portion of these data was used to train the natural language and segment-based speech recognition components. The system obtained an overall understanding accuracy of 52% on the test data, which is similar to our earlier reported results for English [1].

INTRODUCTION

In the fall of 1989, our group first demonstrated VOYAGER, a system that can engage in verbal dialogues with users about a geographical region within Cambridge, Massachusetts [2]. The system can provide users with information about distances, travel times, or directions between objects located within this area (e.g., restaurants, hotels, post offices, subway stops), as well as information such as addresses or telephone numbers of the objects themselves. While VOYAGER is constrained both in its capabilities and domain of knowledge, it contains all the essential components of a spoken-language system, including discourse maintenance and language generation. The VOYAGER application provided us with our first experience with the development of spoken language systems, helped us understand the issues related to this endeavor, and provided a framework for our subsequent system development efforts [3, 4].

Over the past few years, we have become increasingly interested in developing multilingual spoken language systems. There are several ongoing international spoken language *translation* projects whose goal is to enable humans to communicate with each other in their native tongues [5, 6]. Our objective, however, is somewhat different. Specifically, we are interested in developing multilingual human-*computer* interfaces, such that the information stored in the database can be accessed and received in multiple spoken languages. We believe that there is great utility in having such systems, since information is fast becoming globally accessible. Furthermore, we suspect that this type of multilingual system may be easier to develop than speech translation systems, since the system only needs to anticipate the diversity of one side of the conversation, i.e., the human side. During the past year, we have begun to develop a multilingual version of VOYAGER. This paper will describe our work in extending VOYAGER's capability from English to Japanese.

Since VOYAGER was originally designed only for English, a number of changes were necessary to accommodate multiple languages. In the next section, we describe our approach to developing multilingual systems, and the modifications made to the original system. A discussion of the specific implementation of the various components for Japanese will follow. Finally, performance evaluation of the Japanese VOYAGER system will be presented, followed by a brief description of future plans.

SYSTEM DESCRIPTION

Figure 1 shows a block diagram of a prototypical MIT spoken language system. The speech signal is converted to words using our SUMMIT segment-based speech recognition system [7]. Language understanding makes use of TINA, a probabilistic natural language system that interleaves syntactic and semantic information in the parse tree [8]. Data exchange between SUMMIT and TINA is currently achieved via an N-best interface, in which the recognizer produces the top-N sentence hypotheses, and TINA screens them for syntactic and semantic well-formedness within the domain [1]. The parse-tree produced

[1] This research was supported by DARPA under Contract N00014-89-J-1332, monitored through the Office of Naval Research.

[2] Currently a visiting scientist from NEC Corp, Kawasaki, Japan.

[3] The authors are listed in alphabetical order.

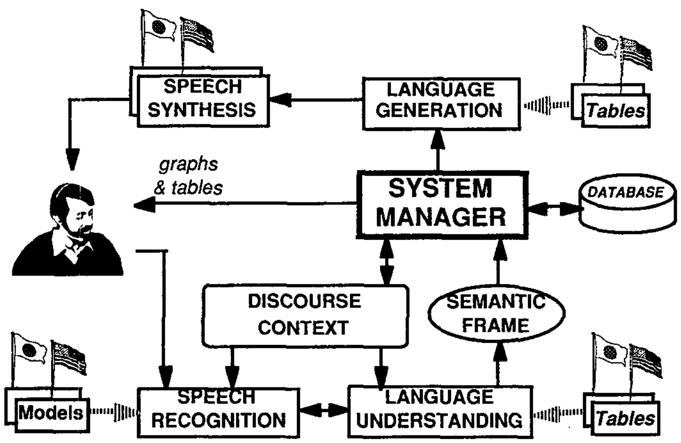

Figure 1: Schematic of prototypical MIT spoken-language system.

by TINA is subsequently converted to a *semantic frame* which is intended to capture the meaning of the input utterance in a language *independent* form [4].

The semantic frame is passed to the system manager which uses it, along with contextual information stored in the discourse component, to access information stored in the database, and provide a response [2]. The VOYAGER application uses an object-oriented database, although we have also accessed data in SQL and other configurations [3]. Responses to the user consist of displays, text, and synthetic speech. The latter two are derived via a language generation component which generates nounphrases from the internal semantic representation and embeds them into context-dependent messages.

In order to develop a multilingual capability for our spoken language systems, we have adopted the approach that each component in the system be as language transparent as possible. In the VOYAGER system for instance, the system manager, discourse component, and the database are all structured so as to be independent of the input or output language. Where language-dependent information is required we have attempted to isolate it in the form of external tables or rules, as illustrated in Figure 1 for both the language understanding and generation components. As will be described in more detail in the next section, we trained a version of the basic SUMMIT system for both Japanese and English, using data recorded from native speakers for each language. The current user interface is very similar to that of the original VOYAGER system, except that a separate recording icon is used for each language. For text-to-speech synthesis we use a DECtalk system for English, and an NEC text-to-speech system for Japanese.

If we are to attain a multilingual capability within a single system framework, the task of porting to a new language should involve only adapting existing tables or models, without requiring any modification of the individual components. By incrementally porting the system to new languages we hope to slowly generalize the architecture of each component to achieve this result. The following sections provide more detailed descriptions of the work done in the different areas to achieve a bilingual status of VOYAGER.

JAPANESE IMPLEMENTATION

To allow VOYAGER to converse with a user in Japanese, the following steps were taken. We first converted the system so that it could generate responses in Japanese. This enabled us to collect data from native speakers of Japanese in a *wizard* mode whereby an experimentor would *translate* the subjects' spoken input and type the resulting English queries to the system [3, 9]. Once data were available we were able to port the speech recognition and language understanding components. In the process of augmenting the system components to handle Japanese, we made many changes to the system core structure, separating out the language-dependent aspects into external tables and rules.

Data Collection

One of the most time-consuming aspects of the porting process was the acquisition of appropriate user data capturing the many different ways users can ask questions within the VOYAGER domain. We started with translations from available English sentences, but these alone are not nearly adequate for closure on coverage of actual data. Although in theory a grammar developer can use his/her innate knowledge of the language to write appropriate grammar rules, in practice such an approach falls far short of complete coverage of actual user utterances.

For data collection from Japanese subjects we recorded data from 40 native speakers, recruited from the general MIT community. In a manner similar to data collection techniques used for the ATIS domain [3], subjects were asked to solve four problem scenarios. At the end of the session subjects were also allowed to ask random questions of the system. The resulting corpus of 1426 utterances was partitioned into a 34 speaker training set and a 6 speaker test set which was subsequently used to evaluate system components.

Speech Recognition

Major tasks in porting SUMMIT to Japanese include acoustic-phonetic, lexical-phonological and language modeling. In an earlier paper, we described these components and reported on a speaker-dependent evaluation [10]. Will briefly summarize our previous work, and describe all subsequent developments, including improved language modeling and speaker-independent training.

Phonetic Modeling In the current version, we use a context-independent mixture (up to 16) diagonal Gaussian model to represent each label in the lexical network [7]. Starting from seed models, the phonetic models are iteratively trained using a segmental K-means-like procedure whereby the forced alignments of the previous iteration are used to train the current iteration. In the English version, the seed models were trained from the manually-aligned phonetic transcriptions of the TIMIT corpus [11]. Rather than obtaining aligned phonetic transcriptions for a Japanese corpus, we found that we could achieve reasonable initial alignments by seeding our Japanese phonetic models from their phonetically most similar English counterparts. Based on an inspection of the alignments, we confirmed that the resulting Japanese models were converging to the intended labels after a few training iterations.

Phonological Modeling Words in the lexicon must be mapped from the abstract phonemic representation to the possible acoustic realizations, taking into account contextual variations. We have adopted the procedure of modeling some of these variations through a set of phonological transformation rules, some of which are unique to Japanese. One of the typical phonological effects that we must account for in Japanese is the different phonetic realizations of the so-called mora (syllabic) phonemes /Q/ and /N/. For example, the phoneme /Q/ is regarded to occupy one higher-level temporal unit (mora) and is realized as a lengthening of the closure interval before stop consonants. When it is followed by fricatives, it may be realized instead as a lengthening of the following frication. Another major phonological phenomenon is the devoicing of /i/ and /u/, which typically occurs when they are preceded and followed by voiceless consonants.

In the English version of SUMMIT, phonological transformation rules have been used to generate alternative pronunciations based on low-level phonological effects such as flapping, palatalization, and gemination. For the Japanese version, we have been able to use the same framework for the conversion of mora phonemes into different phonetic realizations as well as describing lower-level phonological effects such as gemination and devocalization. A set of approximately 60 phonological rules has been developed to account for the possible acoustic realizations of word sequences. These rules produce a total of 56 distinct acoustic labels in the resulting lexical network.

Language Modeling Language modeling is an important aspect of speech recognition since it can dramatically reduce the difficulty of a task. Many speech recognition systems developed for English, particularly those developed for spontaneous speech, employ n-gram language models which capture local word constraints in an utterance [4, 12]. On the other hand, most speech recognition systems for Japanese speech currently employ only small and rather constrained context-free grammars which may not be well suited to spontaneous speech [13].

Word ID	Pronunciation	Left Category	Right Category
ta	t a	aux-tai	adj-r
tara	t a r a	aux-tara	aux-tara
Q	q	inf-v-soku	v-p-soku
te	t e	aux-te	aux-te
de	d e	p-c-de	p-c-de
desu	d e s u	aux-desu	aux-desu-f
to	t o	p-c-to	p-c-to
to(p-j)	t o	p-j-to	p-j-to

Figure 2: Example lexical entries. Each lexical entry consists of a word ID, a pronunciation, and left and right morphological categories.

Compared to English, the choice of lexical units for Japanese speech recognition is less clear. In particular, Japanese orthography does not have spacing between words, making it difficult to have a common agreement on where word boundaries are in a sentence, especially in the case of certain function word sequences. The choice of units impacts both the compactness of the lexical representation and the effectiveness of local grammatical constraints. If we choose units that are too large, the lexicon will need many redundant entries to capture the linguistic variation. On the other hand, choosing smaller units weakens the constraint available from local language models such as statistical bigrams. We have addressed this to some degree by carefully choosing a set of morphological units along with left and right adjacency categories for these units. For example, lexical entries are fully separated into root and inflectional suffixes, except for words with irregular inflections, thus providing a system flexible enough to cope with various expressions in spontaneous speech.

In order to develop sufficiently general grammatical constraints to be used for continuous speech recognition, we developed a category bigram grammar, where the classes are defined by morphological categories. As illustrated in Figure 2, each lexical entry is given a left and right morphological adjacency category. The probability of the word w_j given word w_i is defined to be

$$\begin{aligned} p(w_j|w_i) &\approx \hat{p}(l(w_j)|w_i)\ \hat{p}(w_j|l(w_j)) \\ &\approx \hat{p}(l(w_j)|r(w_i))\ \hat{p}(w_j|l(w_j)) \\ \hat{p}(w_j|l(w_j)) &= \frac{1}{L(l(w_j))} \end{aligned}$$

where $l(w)$ and $r(w)$ are the categories of word w as viewed from the left and right respectively, and $L(l)$ is the number of distinct words in a category l. By this definition, all words within a category are assumed to be equally probable.

As we and others have done previously [4, 12], the category bigram probability is smoothed by interpolating the bigram estimate with the prior probabilities of each category:

$$\hat{p}(l|r) = \lambda(r)\,\frac{c(r,l)}{c(r)} + (1-\lambda(r))\,\frac{c(l)}{c(all\ word\ tokens)}$$

$$\lambda(r) = \frac{c(r)}{c(r)+K}$$

where $c(x)$ is the count of tokens of category x in the corpus.

Language Understanding

The grammar for the English VOYAGER had been entered in the form of context-free rules plus constraints. A trace mechanism was used to handle movement phenomena, and syntactic and semantic features were unified during parsing to invoke agreement constraints. Japanese was in many respects easier than English – we found that it was unnecessary to mark any syntactic or semantic features, and Japanese, unlike European languages, appears not to make use of constituent movement. The only difficulty with Japanese was that parse trees tend to be left-recursive, which can cause infinite-loop problems in a top-down parser. Noun phrase modifiers are positioned to the left of the modified object, and, furthermore, the preposition indicating the relationship *follows* the modifier. Thus a top-down depth-first parser can keep seeking a noun modifier as the next constituent, at the end of an infinite series of recursive modifiers.

Since the main reason for parsing top-down was the trace mechanism, which Japanese does not use, our solution was to implement a simple bottom-up parser without trace. Rules were entered by hand, based on all of the training material we had collected. Figure 3 shows an example parse for the sentence, *"Sentoraru Eki no chikaku no toshokan wa doko desu ka?"* (*"Where is a library in the vicinity of Central Station?"*). The left-recursion is apparent from the shape of the parse tree, and the potential for infinite recursion is clear from the category labels on the left-most branch, since "A-PLACE" can rewrite as ("A-PLACE" ...).

Meaning Representation

The Japanese parse tree must be converted to a semantic representation in order to access the information in the VOYAGER knowledge base. To do this, we designed the grammar rules for the Japanese grammar such that the resulting parse tree could easily be converted to a semantic frame essentially identical to that of the corresponding English sentence. A table-driven procedure is used to convert the parse tree to the semantic frame for both languages. The functions that carry out the conversion are essentially language independent, with the language-dependent information being stored in separate tables for each language. We have found that the original semantic frame designed for English can accommodate Japanese with only minor modifications.

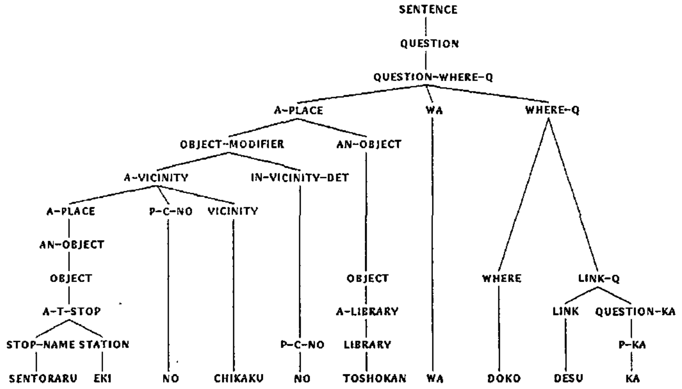

Figure 3: Parse tree for the sentence, *"Sentoraru Eki no chikaku no toshokan wa doko desu ka?"* (*"Where is a library in the vicinity of Central Station?"*).

Semantic Associations for Relevant Parse Nodes:

Parse Category	Semantic Category	Function
question-where-q	locate	set-sentype
a-t-stop	station	noun-phrase
library	public-building	noun-phrase
stop-name	stop-name	proper-name
vicinity	j-near	j-operator

Terminal Translations:

toshokan	library
Sentoraru	central

Table 1: Control tables required to convert from parse tree of Figure 3 to semantic frame of Figure 4. These include mappings from parse tree categories to semantic categories to functional types, as well as translations for critical content words.

Given a well-constructed grammar, it is a relatively simple process to define the conversions from a parse tree to a semantic frame. Semantic encoding is defined at the level of the grammatical category, identified with each node in the parse tree, rather than at the level of an entire rule. All of the semantic encoding instructions are entered in the form of simple association lists. Each semantically active category (preterminal or nonterminal) in the parse tree is associated with a corresponding semantic name, which is often the same as its given name. Each unique semantic name is in turn associated with a functional type, defining what function to call when this node is encountered in the parse tree during the stage of converting the parse to a semantic frame. There are fewer than twenty distinct functional types.

The function that converts a parse tree to a semantic frame visits each node once in a top-down left-to-right fashion, calling the appropriate functions as dictated by the mappings. Table 1 gives the complete set of category correspondences required in order to produce a semantic frame from the parse tree in Figure 3. Notice that most of the nodes in the parse tree are ignored. The semantic categories shown in the table are all identical to those for English except for the special category "j-near" corresponding to the function "j-operator," specialized to handle Japanese postpositional particles. The "j-operator" function renames the generic key "topic" in the semantic frame under construction to the specific semantic relationship defined by the particular operator, in our case, "near." In addition to these mappings, a translation table must also be provided for those words that carry semantic information. Only two words in this sentence need to be provided, as shown in the table.

Ultimately, upon complete analysis of a parse tree, a nested semantic frame is produced – a structure with a name, a type, and a set of [key–value] pairs, where the value could be a string, a symbol, a list of values, a number, or another semantic frame. The semantic frame for our example sentence is shown in Figure 4. Entries in the frame are order-independent, and the same semantic frame is produced from a large pool of questions with different phrasings but equivalent meanings, such as "*What is the distance between MIT and Harvard,*" and "*How far is it from MIT to Harvard.*" Likewise, Japanese versions of this question produce a semantic frame that is essentially identical to the one produced for English.

We had anticipated that the very different order of constituents between Japanese and English might make it hard to produce an equivalent semantic frame from a Japanese sentence to that produced by an English sentence with the same meaning. This did not turn out to be the case. Except for the additional special functions to handle post-positional particles, along with a few other minor adaptations, we were able to use the same functional procedures for converting Japanese parse trees to semantic frames as those used for English. By carefully choosing grammar rules with correspondences to their English equivalents, we were able to exploit the same protocol for producing a semantic frame, thus feeding into the main system with a common interlingual representation. We feel that the success of this approach is largely attributable to the fact that we have intentionally designed our semantic interpretation procedure to operate at the level of independent parse tree nodes, rather than to be explicitly associated with grammar rules or with complex patterns found in the parse tree.

System Manager & Discourse Component

The system manager and discourse components attempt to process an input semantic frame in the context of a discourse and provide an appropriate response to the user [2]. Normally this will involve accessing the database for the set of objects satisfying the input constraints, although in the case where a query is ambiguous, some sort of clarification might be appropriate. In the example shown in Figures 3 and 4 for instance, the result would be the set of libraries having the property that they are near a station named "Central". These components are structured so that they are language independent (i.e., the resulting set would be identical no matter what the input language was). The net effect is that the input and output languages are completely isolated from each other so that a user could speak in one language and have the system respond in another. Additionally, 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.

```
(LOCATE CLAUSE
      TOPIC: [library] REFERENCE
            REFTYPE: PUBLIC-BUILDING
            PREDICATE: NEAR  PREDICATE
                      TOPIC: [central] REFERENCE
                            REFTYPE: STATION)
```

Figure 4: Semantic frame produced by parse tree of Figure 3 using mappings defined in Table 1.

Language Generation

Once the system manager has determined an appropriate response for the user it will display the result on the map, and use the language generation component to produce a verbal answer. The language generation component has the ability to generate noun phrases describing object sets produced by the system manager. The noun phrase can be singular or plural, and can contain a definite or indefinite article. For the example of the set of libraries near Central Station, the English noun-phrase generator could produce "*library near Central Station*", or "*libraries near Central Station*", along with the articles "*a*" or "*the*" depending on the need. These conditions can be specified by the system manager at the moment of generation since the precise context of the response is known.

The noun phrases produced by the generator are embedded in language-dependent message strings which are stored in a table. Each string is given a unique label so it can be referenced by the system manager. Each language thus requires an association list of the message label and string pattern. To produce a response, the system manager calls the language generation component with a particular message label, and the noun phrases associated with the response. In the library example for instance, the system knows of one library near Central Station. It would therefore call the language generation component with an *only* message, and pass as arguments the noun-phrase "*library near Central Station*" or "*Sen-*

toraru Eki no chikaku ni aru toshokan" depending on whether the output language were English or Japanese. The respective unknown messages consist of "*I know of only one <noun-phrase>.*" or "*<noun-phrase> wa hitotsu dake shitte imasu.*"

Although the language generation process has been presented as a two-stage process, it is actually recursive since as is the case for our example, a noun-phrase can itself consist of many embedded noun-phrases. To build up the noun-phrase for the set of libraries near Central Station, the generator would start with the basic vocabulary value for library, and embed this string using the *near* message and the string value of the noun-phrase Central Station. In English, the near message would be of the form "*<noun> near <object>*". Using this procedure, the language generation component can create arbitrarily complicated noun-phrases in the domain.

EVALUATION

For the Japanese VOYAGER system, we defined a vocabulary of 495 words comprised of words in the training set and words determined by translating 2000 sentences from the English VOYAGER training corpus. This vocabulary covered 99% of the words in the test set (96% of unique words). The category bigram was also trained using the training data and had perplexities of 25.9 and 27.5 on the training and test sets respectively. First choice word and sentence error rates were 14.9% and 53.3%, respectively, on the test set.

The parser covers 82% percent of the training data, and 65% of the test data. An inspection of the answers generated by the system using text input showed that 60% of the responses for the test set was correct. The performance of the system dropped by 8%, to 52%, when the input is spoken rather than typed ($N = 10$ for the N-best interface). Note that the system's understanding ability actually exceeds its sentence recognition accuracy by 5%, which suggests that a full transcription is not always necessary for understanding. Finally, this performance is similar to that initially reported for our English system when using context-independent phone models with a word-pair grammar of similar perplexity (22) [1].

FUTURE PLANS

In this paper we described our recent effort at converting VOYAGER to a bilingual platform. We are encouraged by our preliminary results, and will continue to improve its capabilities in all directions, including context-dependent phonetic models, a robust parsing capability modeled after our ATIS system, and an expansion of its knowledge domain. We are currently porting the VOYAGER system to other languages including French, Italian, and German. We plan to collect data for all languages in scenario collection format in order to acquire more goal-oriented speech. We would also like to incorporate a pointing mechanism into the system, since the VOYAGER application lends itself to this kind of multimodal input.

REFERENCES

[1] Zue, V., Glass, J., Goddeau, D., Goodine, D., Leung, H., McCandless, M., Phillips, M., Polfroni, J., Seneff, S., and Whitney, D., "Recent Progress on the MIT VOYAGER Spoken Language System," *Proc. ICSLP*, 1317–1320, Kobe, Japan, November 1990.

[2] Zue, V., Glass, J., Goodine, D., Leung, H., Phillips, M., Polifroni, J. and Seneff, S. "Preliminary Evaluation of the VOYAGER Spoken Language System," *Proc. DARPA Speech and NL Workshop*, 160–167, Harwichport, MA, October 1989.

[3] Seneff, S., Glass, J., Goddeau, D., Goodine, D., Hirschman, L., Leung, H., Phillips, M., Polifroni, J., and Zue, V., "Development and Preliminary Evaluation of the MIT ATIS System," *Proc. DARPA Speech and NL Workshop*, 88–93, Pacific Grove, CA., February 1991.

[4] Zue, V., Glass, J., Goddeau, D., Goodine, D., Hirschman, L., Phillips, M., Polfroni, J., Seneff, S. "The MIT ATIS System: February 1992 Progress Report," *Proc. DARPA Speech and NL Workshop*, 84–88, Harriman, NY, February 1992.

[5] Roe, D. B., Pereira, F., Sproat, R. W., Riley, M. D., Moreno, P. J., and Macarron, A., "Toward a Spoken Language Translator for Restricted-domain Context-free Languages," *Proc. Eurospeech*, 1063-1066, Genova, Italy, September 1991.

[6] Morimoto, T., Takezawa, T., Ohkura, K., Nagata, M., Yato, F., Sagayama, S., and Kurematsu, A., "Enhancement of ATR's Spoken Language Translation System: SL-TRANS2," *Proc. ICSLP*, 397–400, Banff, Canada, October 1992.

[7] Phillips, M., Glass, J., and Zue, V., "Automatic Learning of Lexical Representations for Sub-Word Unit Based Speech Recognition Systems," *Proc. Eurospeech*, 577–580, Genova, Italy, September 1991.

[8] Seneff, S., "TINA: A Natural Language System for Spoken Language Applications," *Computational Linguistics*, Vol. 18, No. 1, 61–86, 1992.

[9] Zue, V., Daly, N., Glass, J., Goodine, D., Leung, H., Phillips, M., Polifroni, J., Seneff, S. and Soclof, M. "The Collection and Preliminary Analysis of a Spontaneous Speech Database," *Proc. DARPA Speech and NL Workshop*: 126–134, October 1989.

[10] Sakai, S., and Phillips, M., "J-SUMMIT: A Japanese Segment-Based Speech Recognition System,", *Proc. ICSLP*, 1515–1518, Banff, Alberta, Canada, October 1992.

[11] Lamel, L., Kassel, R., and Seneff, S., "Speech Database Development: Design and Analysis of the Acoustic-Phonetic Corpus," *Proc. DARPA Speech Recognition Workshop*, Report No. SAIC-86/1546, 100–109, February 1986.

[12] Kubala F., et al. "BBN BYBLOS and HARC February 1992 ATIS Benchmark Results", *Proc. DARPA Speech and NL Workshop*, 72–77, Harriman, NY, February 1992.

[13] Itou, K., Hayamizu, S., Tanaka, H., "Continuous Speech Recognition by Context-Dependent Phonetic HMM and an Efficient Algorithm for Finding N-best Sentence Hypotheses," *Proc. ICASSP*, 21–24, San Francisco, CA, March 1992.