MULTILINGUAL SPOKEN-LANGUAGE UNDERSTANDING IN THE MIT VOYAGER SYSTEM

James Glass, Giovanni Flammia, David Goodine, Michael Phillips,
Joseph Polifroni, Shinsuke Sakai, Stephanie Seneff, and Victor Zue[1]

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

ABSTRACT

This paper describes our recent work in developing multilingual spoken language systems that support human-computer interactions. Our approach is based on the premise that a common semantic representation can be extracted from the input for all languages, at least within the context of restricted domains. In our design of such systems, language dependent information is separated from the system kernel as much as possible, and encoded in external data structures. The internal system manager, discourse and dialogue component, and database are all maintained in a language transparent form. Our description will focus on the development of the multilingual MIT VOYAGER spoken language system, which can engage in verbal dialogues with users about a geographical region within Cambridge, Massachusetts in the USA. The system can provide information about distances, travel times, or directions between objects located within this area (e.g., restaurants, hotels, banks, libraries), as well as information such as the addresses, telephone numbers, or location of the objects themselves. VOYAGER has been fully ported to Japanese and Italian, and we are in the process of porting to French and German as well. Evaluations for the English, Japanese and Italian systems are reported. Other related multilingual research activities are also briefly mentioned.

Keywords: spoken-language systems, speech understanding, multilingual, spontaneous speech

1. INTRODUCTION

Computers are fast becoming a ubiquitous part of our lives, and people's appetite for information is ever increasing. As a result, it is essential that we seriously address the issue of finding naturally accessible interfaces so that a majority of the population can access,

[1] Names in alphabetical order after the first author. Shinsuke Sakai participated in this work while he was a visiting scientist on leave from NEC Corporation. He currently works in NEC's Human Language Research Laboratory, Information Technology Research Laboratories. David Goodine now works at NYNEX Science and Technology, Inc. Mike Phillips now works at Applied Language Technologies, Inc.

process, and manipulate vast amounts of information for education, decision-making, purchasing, and entertainment. A speech interface, in a user's own language, is highly desirable because it is natural, flexible, efficient, and an economical form of communication among humans.

When people think of speech input to computers, the problem that immediately comes to mind is automatic speech recognition. Speech recognition is a very challenging problem in its own right, with well-defined applications such as dictation, transcription, and simple data entry. However, a preponderance of tasks appropriate for spoken input fall into the realm of problem solving. In these applications, such as information retrieval and interactive transactions, the solution is often built up incrementally, with the user and the computer both playing an active role in the conversation. To achieve this goal, several language-based technologies must be developed and integrated. On the input side, speech recognition must be combined with natural language processing in order to derive an *understanding* of the spoken input, often in the context of previous parts of the verbal dialogue. On the output side, some of the information that the user seeks as well as any clarification dialogue generated by the system must be converted to natural sentences and, possibly, delivered as verbal responses.

In the last few years, a number of spoken language systems have appeared in the research community, in which a user can typically carry on a spoken dialogue with a computer in order to retrieve information from a database, within a narrow domain of expertise. Most of these systems involve accessing time schedule information for transportation services such as trains (Clementino and Fissore, 1993; Eckert et al., 1993; Oerder and Aust, 1994), airplanes (Peckham, 1991) or ferries (Blomberg et al., 1993). In the United States, a number of different sites in the ARPA program have also developed speech understanding systems in the air-travel domain (Pallett et al., 1994).

Our group's involvement with spoken language system development started in late 1989, when we first demonstrated the VOYAGER system (Zue et al., 1989b). As shown in Figure 1, VOYAGER can engage in verbal dialogues with users about a geographical region within Cambridge, Massachusetts, in the USA. It 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 nevertheless 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 spoken language system development efforts (Seneff et al., 1991; Zue et al., 1992; Zue et al., 1993; Goddeau et al., 1994).

Recently, 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 one another in their native tongues (Waibel et al., 1991; Roe et al., 1991; Morimoto et al., 1993; Wahlster, 1993) 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, and the topic of conversation is typically quite focused.

We have found the VOYAGER system to be a particularly attractive vehicle for exploring multilinguality. As of the present time, VOYAGER operates in a trilingual mode, where the user can select among the three choices, English, Japanese, or Italian, for the communication language. A user can also freely mix the three languages in a single conversation, and the system will incorporate context appropriately, regardless of the language of the context-setting query(s). We have collected a large number of training sentences for all three of these languages, and we have evaluated VOYAGER for independent test sets for each of the three languages.

This paper starts with a description of our approach to developing multilingual spoken language systems in support of human-computer interactions. We will then describe the actual implementation of this approach in the VOYAGER domain, focusing on the porting of the original English version to Japanese and Italian, with occasional references to ongoing efforts in porting the system to French and German. We will end by briefly discussing our other related activities in multilingual research, and summarizing our future plans.

2. GENERAL DESCRIPTION

2.1 System Architecture

Figure 2 shows a prototypical architecture of a spoken language system developed in our group, emphasizing its multilingual nature. The language-independent aspects of the system components are described in more detail in the following sections. This is followed by a discussion of issues raised in porting the system to new languages.

2.1.1 Speech Recognition

The speech signal is first converted to word hypotheses using a segment-based speech recognizer we have developed in our group called SUMMIT. SUMMIT begins the recognition process by locating acoustic-phonetic landmarks in the speech signal, and subsequently creating an acoustic-phonetic network (Zue et al., 1989a). In the VOYAGER domain, phonetic units are modelled with a context-independent mixture of up to 16 diagonal Gaussians (Phillips et al., 1991). Baseform phonemic pronunciations in the lexicon are expanded using a set of phonological rules to create a pronunciation network. Recognition is performed using an A* search by matching the acoustic-phonetic network with the pronunciation network to produce either N-best sentence or word-graph outputs (Zue et al., 1990b; Hetherington et al., 1993). A bigram language model is typically used at the initial search stage, although early versions of the VOYAGER system used word-pair language models (Zue et al., 1989b).

2.1.2 Language Understanding

The language understanding component makes use of a probabilistic natural language system we have developed in our group called TINA (Seneff, 1992). TINA is based on a context-free grammar augmented with a set of features used to enforce syntactic and semantic constraints, and including a trace mechanism to handle movement phenomena. The grammar for a given language is written by hand, and the rules are converted to a set of sibling-sibling transition probabilities conditioned on the parent[2]. Essentially, the rules are broken apart into a set of trigram probabilities capturing both spacial (parent) and temporal (left-sibling) conditioning contexts. The probabilities are trained by tabulating counts in parse trees obtained automatically from a large set of training sentences. The context-free rules contain no information about agreement constraints. Instead, features are associated with the *category names*, and unifications apply based only on the category, not on the rule. In fact, the rules are not represented explicitly anywhere in the grammar or the parsing process. For example, the `subject` category typically unifies the feature `number` between its left-sibling and its children. Terminal words that have feature values unify them with whatever feature pattern is delivered to them by their parent/left-sibling during the parse process. Certain categories can *isolate* their descendants from their siblings by erasing the feature values before transitioning to the children. This is a useful mechanism for isolating individual clauses, for example, or for prohibiting agreement between verbs and their objects. Syntactic features enforce constraints such as subject-verb agreement, and semantic features are particularly important for constraining gaps. An example parse tree for the sentence, "Where is the library near Central Square?" is shown in Figure 3.

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-formed-ness within the domain (Zue et al., 1990b).

2.1.3 Meaning Representation

If an N-best sentence is parsable by TINA, the resulting parse tree is converted to a *semantic frame* which is intended to capture the meaning of the input utterance in a language-independent form. Through our previous experience in developing spoken language systems, we have learned that simplicity of form is an important principle in building effective meaning representations. Our view on the appropriate structural units of a semantic frame has evolved over time. Our present view is that all major constituents in a sentence can be classified as one of only three distinct categories, which we label as [`clause`], [`topic`], and [`predicate`]. Thus, verbs, adjectives, prepositions and modifier nouns are all considered to be predicates. Furthermore, constituencies such as "subject" and "direct object" are not explicitly marked. Instead, the role of a `[topic]` is inferred from its position in the hierar-

[2] left-hand-side of the set of rules sharing the same left-hand category.

chical structure. An example semantic frame for the sentence, "Where is the library near Central Square," is shown in Figure 4.

The process of converting a parse tree to a semantic frame is straightforward. A mapping is specified at the *category* level, rather than at the *rule* level or some more complex parse-tree pattern specification. This permits much more generality across languages, where the *ordering* of constituents may vary greatly but the *hierarchy* usually is maintained (i.e., subjects and predicates are siblings within clauses; prepositional phrases are nested inside the noun phrases they modify, whether left- or right-attached). The choice of category names in the parse tree reflects both syntactic and semantic roles, and is carefully selected. Many of the parse-tree categories carry no semantic role; these are simply left out of the mapping table. The mapping is generally many-to-one: several different parse-tree categories may map to the same semantic-tree category. By simply replacing the names of each active parse-tree category with the appropriate semantic association, an individual parse tree is converted into a *semantic tree.* Figure 5 shows the semantic tree corresponding to the parse tree of Figure 3.

Each unique semantic-tree category is assigned a particular *structural* role. In VOYAGER there are fewer than twenty unique roles. These include labels such as `predicate, quantifier, numeric, superlative, conjunction,` etc. Once the semantic tree is assigned, a recursive routine carries a *semantic frame* under development through the semantic tree in a top-down, left-to-right fashion, using the assigned structural role of each semantic category to decide both the assignment of the constituents of the semantic frame and which components of the semantic frame to pass along to the children. A complete semantic frame emerges at the termination of the treewalk.

The semantic frame serves many roles in our spoken language systems: it is used as the basis for accessing information from application databases, to maintain a discourse history, and also for natural language generation. Since the frames are intended to capture the relevant semantic information of the input query, they can also be used to paraphrase the input. This latter capability has proved to be quite useful for multilingual development.

2.1.4 System Manager and Discourse Component

The System Manager uses the semantic frame, along with contextual information stored in the discourse component, to access information stored in the database and provide a response (Zue et al., 1989b). The VOYAGER application stores information about objects in a simple database, although we have also accessed data in relational databases in other configurations (Seneff et al., 1991; Zue et al., 1993; Goddeau et al., 1994). The current VOYAGER database contains information on approximately 150 objects.

The discourse capabilities of the VOYAGER system are simplistic but nonetheless effective in handling the majority of interactions within its domain. The discourse component reserves two slots for anaphora resolution. The first slot refers to the location of the user, while the second refers to the most recently referenced set of objects. This information enables the system to determine the correct action to questions such as "What is their address?" and "How far is it from here?".

The discourse component also has a simple mechanism for handling ambiguous queries where not enough information has been provided by the user. Examples of such queries would be "How far is a bank?", since there are many banks, or "How do I get to MIT?", if there is no reference to compute directions from. The system handles such ambiguities by pointing out the problem to the user, and pushing the query onto a stack of incompletely specified queries. When the user provides additional information that is successfully resolved, the top query in the stack is popped for reevaluation. If the additional information is not sufficient to resolve the original query, it is pushed back onto the stack with the new information incorporated. In the case where the clarification is also ambiguous, it is pushed onto the stack as well, until it can be clarified. A protection mechanism automatically clears the history stack whenever the user decides to abandon a line of questioning before all ambiguous queries on the stack are resolved.

2.1.5 Response Generation

Responses to the user consist of graphic displays locating the objects, or directions of interest, as well as a textual and spoken response summarizing the information. The text information produced for the latter two outputs are derived via a language generation component we have developed in our group called GENESIS (Glass et al., 1994). This system produces phrases from the internal semantic representation and embeds them into context-dependent messages. This component has evolved considerably from early English versions of the VOYAGER system, due to our multilingual development. It will be described in more detail later in this paper.

To date we have not actively developed a text-to-speech capability and have used available speech synthesizers to produce spoken output for the user.

2.2 Multilingual Issues

Our approach to developing multilingual spoken language 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 (Hutchins and Somers, 1992). 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. Currently in VOYAGER, for instance, the System Manager, discourse component, and the database 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.

Where language-dependent information is required, we have attempted to isolate it in the form of external models, tables, or rules, as illustrated in Figure 2 for the speech recognition, language understanding, and generation components. Figures 6–8 show parse trees for a sentence in Japanese, Italian, and French, respectively. They, as well as the English version shown in Figure 3, are all derived using the same natural language system, TINA, but with different grammar rules. All these sentences arrive at the same semantic frame shown in Figure 4[3]. For speech recognition, we trained the basic SUMMIT system for the languages of interest, using data recorded from native speakers for each language. For text-to-speech synthesis we acquire an appropriate text-to-speech system for each language.

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.

3. MULTILINGUAL IMPLEMENTATION

To port a spoken language system to another language, the following steps must be taken:

1. *Language Generation:* The system must first be able to generate the appropriate responses in the target language from semantic frames, which are derived from a set of English training sentences.

2. *Data Collection:* A set of sentences in the target language must be obtained for system development, training, and evaluation.

3. *Language Understanding:* Using the collected sentences, a grammar for the target language must be written, and the capabilities of the natural language component must be extended.

4. *Speech Recognition:* Lexical items (with associated pronunciations), acoustic models, and language models must be derived from the training sentences in order to bring up the recognizer in the target language.

5. *Performance Evaluation:* The performance of the system must be evaluated using previously unseen data.

6. *System Refinement:* The capabilities of the system will be improved and refined as more training data are acquired.

In the following sections, we will describe our experience in porting VOYAGER from English to other languages, with most of the discussion focused on Japanese and Italian (Glass et

[3] The only exception is that the quantifier is absent in the Japanese version.

al., 1993; Flammia et al., 1994).

3.1 Language Generation

As shown in Figure 2, the GENESIS language generation component of our conversational systems is controlled by the System Manager. In this capacity its role is to provide answers, clarification requests, help, and other computer-initiated feedback, in order to enable a user-friendly dialogue. Responses are typically constructed from a synthesis of information provided by the user and by the database. The System Manager creates a response semantic frame derived from the input frame and modified to reflect the outcome of the database query. For instance, in VOYAGER, the quantifier and number of the response frame depend on whether the result is a null set ("There are no <NP>"), a single item ("There is only one <NP>"), or a larger set ("There are six <NP>"). The noun phrase <NP> is generated directly from the main topic of the input semantic frame, and may be a complex noun phrase such as "Chinese restaurants on Main Street near a subway stop." This communication serves a useful role in verifying the system's understanding of the input query.

The GENESIS system is composed of three modules: a lexicon, a set of message templates, and a set of rewrite rules (Glass et al., 1994). These modules are language-dependent and external to the system itself. In this way, porting the language generation component of an entire conversational system to a new language is confined to developing a new lexicon, messages, and rewrite rules, with the system kernel remaining the same. Since the semantic frame uses English as its specification language, entries in all lexicons, including English, contain words and concepts found in the semantic frame, expressed in English, with corresponding surface realization forms in the target language. The following sections describe each of these modules in greater detail and provide examples of their use in English and French.

3.1.1 Lexicon

The lexicon's main role is to specify the surface form of a semantic frame entry, including the construction of inflectional endings (gender, case, number, etc.). A sample lexicon for English and French is shown in Table 1. As can be seen in the table, each entry in the lexicon contains a *part of speech* tag (e.g., N (Noun), V6 (Regular Verb #6)), a stem, and various derived forms (e.g., the entry for "which" has several realizations in French depending on gender and number). For entries whose morphological variants are regular there are default endings specified under generic *part of speech* entries (e.g., a typical noun (N) in English forms plurals by the addition of an "s"). These defaults can be overridden by an exception specified for the particular entry, as in the English verbs "be" and "do".

Individual entries in the lexicon are also able to specify grammatical attributes that are necessary to control lexical form. In French for example, nouns can specify their gender (e.g., "flight" is masculine), which is required for proper generation of adjectives and quantifiers. In addition, entries can specify default quantifiers (e.g., "royal_east", a proper noun, prefers a definite article). Furthermore, certain auxiliary verbs, such as "do" and "will", can set the verb mode for the main verb (e.g., to "root" mode in English). Finally, the surface form of a particular lexical entry can be controlled by the semantic class of its parent. For instance, numbers can be entered in the semantic frame as simple integers, and realized as cardinal or ordinal (e.g., "second" vs "two") depending upon the semantic class of their parent.

3.1.2 Messages

The catalog of message templates is primarily used to recursively construct phrases describing the topics, predicates, and clauses of a semantic frame. Table 2 shows example message templates for English and French. A message template consists of a message *name* (e.g., *existential*) and a sequence of one or more word strings and keywords. There is also a mechanism for optionally specifying a default value in the event the keyword has no value (e.g., (:AUX be) uses the verb "be" as a default if there is no value for the :AUX keyword in the semantic frame). The set of message templates controls the ordering of constituents, which are instantiated recursively.

3.1.3 Rewrite Rules

The rewrite-rules are intended to capture surface phonotactic constraints and contractions. For example, in French the sequence "de le" is realized as "du". In English we use rewrite rules to generate the proper form of the indefinite articles "a" or "an", or to merge "a other" into "another".

3.1.4 Paraphrasing Semantic Frames

The language generation mechanism operates by processing pieces of a semantic frame (e.g., a topic, predicate, or clause) and embedding them in context-dependent messages. Although the strings are primarily used to generate a response to the user, we have also found this procedure useful during system development. This is because we are able to produce clause-level strings, which in effect are a *paraphrase* of the input query. We have found the ability to paraphrase a semantic frame to be very useful for a number of different purposes. It serves as a kind of translation among the various languages supported by the system, with the semantic frame acting as a form of "interlingua." It is also of great use to system developers when porting to a new domain or language, by providing a confirmation that the natural language component successfully parsed the input query *and* generated an appropriate semantic frame. Finally, some aspects of the paraphrasing are used as part of the response generation. The interested reader is referred to a recent paper on this GENESIS system (Glass et al., 1994) for further details about the paraphrasing process.

3.2 Data Collection

Generally, when we begin porting to a new language, the first step is to develop a functional generation component in that language. Once this exists, we can begin collecting data from native speakers of the new language, by having a bilingual wizard quickly translate their queries into English, typing the English equivalent to a “cross-mode” system that has been set up with English input, target-language output. We have also augmented these natural data with utterances that have been derived by translating available English training sentences from the domain.

In our original data collection effort for English VOYAGER, we collected nearly 5,000 spontaneous utterances (along with their read version) from 50 male and 50 female native speakers of American English (Zue et al., 1989c). We did not ask users to solve any particular scenarios, but rather we let them freely explore whatever areas were of interest to them.

For Japanese data collection we recorded data from 40 native speakers recruited from the general community[4]. In a manner similar to data collection techniques used for the ATIS air-travel domain (Polifroni et al., 1991), 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 over 1,400 utterances was partitioned into a 34 speaker training set and a 6 speaker test set which was subsequently used to evaluate system components.

For the Italian version of the system we used a combination of read and spontaneous data collection techniques. In a typical session, the speaker was asked to complete two problem scenarios and then to read a set of prepared sentences. The resulting corpora contained over 1,200 read, and 1,100 spontaneous sentences from 49 native speakers from different Italian regions. On average there were 9.5 words per sentence. These data were divided into a 42 speaker training set and a 7 speaker test set.

3.3 Language Understanding

Once we have a set of sentences appropriate for the domain in a given language, we can begin to write grammar rules and define a vocabulary set so that these sentences can be properly understood. Generally, we work with an arrangement that allows momentary switching back and forth between English and the target language. This permits the grammar developer to look at a parse tree in English for the equivalent English sentence, and to use this as a guide for developing rules for the target language. While the input language is toggled back and forth, the output language is maintained in English, so that a successful parse leads to a semantic frame that shows an English paraphrase of the sentence being analyzed. The developer can use this paraphrase as a debugging tool: if the English paraphrase is correct then the semantic frame is probably also correct. We have thus far found that Italian, French, and German are sufficiently similar to English that they can use the same tables as are used for English (when augmented with a translation dictionary) for mapping parse trees to semantic frames.

We had anticipated that the different order of constituents between languages such as Japanese or Italian when compared to English might make it hard to produce a semantic frame

[4] This population is probably not representative of the general Japanese population, due to their experience of living in the United States.

from a sentence produced in these languages that is equivalent to that produced by an English sentence with the same meaning. This did not turn out to be the case. In Japanese for instance, except for a few minor adaptations, including some additional special functions that had to be written to handle post-positional particles, we were able to use the same functional procedures for converting Japanese parse trees to semantic frames as those used for the European languages, but with a distinct semantic mapping table. We expect that post-positional particles will appear in future languages, so that these augmentations will have a more general applicability.

We were pleased that our mechanism for translating parse trees to semantic frames generalized well to other languages besides English. We believe this success is due in large part to the fact that 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 or associated with some complex patterns found in the parse tree. As discussed in Section 2.1, all of the semantic encoding instructions are entered in the form of simple association lists defining a corresponding semantic name for each active category, which is often the same as its given name. Because the hierarchy of parse tree constituents (but not necessarily the temporal order) is generally consistent across languages, this method works effectively.

3.4 Speech Recognition

Major tasks in porting SUMMIT to a different language include acoustic-phonetic, lexical-phonological and language modeling. This stage of the porting will focus on our work with Japanese and Italian. Since this process has been discussed extensively elsewhere (Sakai and Phillips, 1993; Flammia et al., 1994), it will be briefly summarized here.

3.4.1 Phonetic Modelling

In the VOYAGER system, each acoustic label is modelled with a context-independent mixture of diagonal Gaussians (Phillips et al., 1991). The choice of the number of acoustic labels and mixture size is language and corpus dependent. For English, we used 58 models based on the labels used in the TIMIT corpus (Zue et al., 1990a). The Japanese version used 74 models based on a set of basic phonemes (long, and short vowels) augmented with frequent syllables (Sakai and Phillips, 1993). For Italian, there were 67 models consisting of the basic Italian phones (stressed and unstressed vowels) along with phone labels that modelled the most frequent one and two syllable words in the training set (Flammia et al., 1994). These word-dependent phone models were intended to account for the many inflected forms of function words of Italian.

Starting from seed models, the phonetic models were 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 (Zue et al., 1990a). Rather than obtaining aligned phonetic transcriptions for the Japanese or Italian corpora, we found that we could achieve reasonable initial alignments by seeding the phonetic models from their most phonetically similar English counterparts. Based on an inspection of the alignments, we confirmed that the resulting models were converging to the intended labels after a few training iterations.

3.4.2 Phonological Modelling

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. 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. In our initial implementation for Italian, the phonological rules were contrained primarily to word-internal effects such as the optional insertion of a schwa between a flap and a consonant.

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. 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.

3.4.3 Language Modelling

Language modeling is an important aspect of speech recognition since it can dramatically reduce the difficulty of a task. Many speech recognition systems, particularly those developed for European languages, employ class n-gram language models which capture local word constraints in an utterance (Brown et al., 1992; Kubala et al., 1992). 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 (Itou et al., 1992).

Compared to English or Italian, 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 for Japanese, where the classes are defined by morphological categories. As illustrated in Table 3, 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.

4. EVALUATION

4.1 English Evaluation

The English version of VOYAGER was first evaluated using a vocabulary of 570 words and a word-pair grammar with test-set perplexity of 22 (Zue et al., 1989d). This system achieved test-set word and sentence error-rates of 14.3% and 49.2%, respectively. With a smaller vocabulary of 381 words, and a bigram grammar with test-set perplexity of 9.0, the word error-rate is reduced to 12.6%.

The parser covers 78% of training and 72.5% of the test data. An inspection of the responses produced by the system on the test set shows that when an utterance was successfully parsed it was able to produce the correct action 97% of the time. This is a result of the fact that the coverage of the parser was tied to the capabilities of the system during development. Using an N-best interface with $N = 10$ the system was judged capable of responding correctly to approximately 50% of all input queries in the test data.

4.2 Japanese Evaluation

For the Japanese version of VOYAGER, we defined a vocabulary of 495 words comprised of words in the Japanese training set and words determined by translating 2,000 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% of the orthographical transcriptions 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 were 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 Japanese system's understanding ability actually exceeds its sentence recognition accuracy by 5.3%, which suggests that a full transcription is not always necessary for understanding.

4.3 Italian Evaluation

For Italian the recognizer vocabulary size was 725 words. This increased size compared to the other languages reflects the fact that Italian is a highly inflected language. The bigram language model had perplexities of 12 and 21.9 on the training and test sets. First choice word and sentence error rates were 20.3% and 70.2% respectively, on the test set. The larger error rates for Italian were due partly to the larger vocabulary size with a limited training set. The increased size of the Italian vocabulary due to inflections was also a source of many of the recognizer errors. Although TINA has the capability of handling agreement constraints, such constraints were enforced only minimally, since it was desirable not to reject a sentence due to an inflectional error on the part of the recognizer.

The Italian parser covered 73.7% of the orthographic transcriptions of the training data, and 68.2% of the test data. A native Italian speech researcher who is not a member of our group evaluated subjectively the system responses for the 252 test sentences, for spoken input. She judged 48% of the responses to be correct, and an additional 16% to be partially correct. The system provided no answer for 25% of the queries, and the remaining 11% of the responses were judged to be incorrect. In spite of the fact that only 30% of the queries were recognized perfectly, the system was able to respond correctly nearly half of the time. The system understanding ability exceeded by far its recognition ability, because many of the word substitution errors were not semantically relevant, and the grammar had loose syntactic constraints, as mentioned previously.

5. SUMMARY AND FUTURE PLANS

In this paper we presented our approach to developing multilingual spoken language systems, and described our recent effort at converting VOYAGER to a multilingual 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 (Zue et al., 1992), and an expansion of the knowledge domain in order to focus on large vocabulary speech understanding. In the latter area we have initiated an effort in the general area of travel, using such real knowledge sources as yellow pages information and census maps (Goddeau et al., 1994).

In porting our systems to other languages we have required a corpus collected from native speakers of the language, and a bilingual developer who can understand the mechanisms used for our English system and convert them to the target language. These mechanisms range from baseform representations, phonological rules, and language models for the speech recognition component, to parse rules for the language understanding component, and message tables for the generation component. Based on our experience with Japanese and Italian, we estimate that it takes between 6 months and a year for a bilingual developer to port an existing system from English to the target language.

We are currently porting the VOYAGER system to French and German, and are interested in other languages such as Spanish and Mandarin. We plan to collect data for all languages by requiring subjects to solve specific scenarios in order to acquire more goal-oriented speech. We are also interested in exploring alternative input modalities such as pointing, since the VOYAGER application lends itself to this kind of multi-modal input.

The current user interface of trilingual VOYAGER is very similar to that of the original VOYAGER system, except that a separate recording icon is used for each language. While the system relies on the user to click on the correct icon to select the proper input language, we have found that, for languages that are quite disparate such as English and Japanese, it is possible to run both recognizers and select the one that results in a better recognition score. More recently, we have started to investigate the possibility of automatic language identification, and have developed a segment-based, probabilistic automatic language identification system with performance comparable to that of others in the literature (Hazen and Zue, 1994).

Our experiences in porting from English VOYAGER to Japanese and Italian have led us to believe that, for restricted domains, one can achieve a performance of around 50% understanding of spoken queries even with a relatively small vocabulary, a limited-coverage grammar, and no robust parsing capabilities. To attain a substantially better performance, however, will probably require an extensive expansion of the grammar rule set, as well as a large increase in the amount of training data and, perhaps, a better use of the parser probabilities to constrain the recognizer search. It has not been our intent, with VOYAGER, to *perfect* the system in any particular language; rather, VOYAGER has been viewed as a testbed for addressing a broad spectrum of issues related to language generality and ease of porting. We feel that, beyond a certain base level of about 50% competence, the gains achieved through additional efforts may reach a point of diminishing returns. We also suspect that improved usability can be achieved through greater attention to the dialogue model, including more feedback to the user on the capabilities of the system, and better error recovery mechanisms. These are issues that we are currently exploring within VOYAGER and our other spoken language systems.

ACKNOWLEDGEMENTS

The authors would like to thank the two anonymous reviewers who provided several suggestions which helped improve the overall paper. This work was supported by ARPA under Contract N00014-89-J-1332, monitored through the Office of Naval Research.

REFERENCES

M. Blomberg, R. Carlson, K. Elenius, B. Granstrom, J. Gustafson, S. Hunnicutt, R. Lindell, and L. Neovius (1993) “An Experimental Dialogue System: Waxholm,”, *Proc. Eurospeech-93*, pp. 1867-1870.

P. Brown, V. Della Pietra, P. De Souza, and J. Lai (1992) “Class-based *N*-gram Models of Natural Language,” *Computational Linguistics*, Vol. 18, no. 4, pp. 467–479.

D. Clementino and L. Fissore (1993) “A Man-machine Dialogue System for Speech Access to Train Timetable Information,” *Proc. Eurospeech-93*, pp. 1863-1866.

W. Eckert, T. Kuhn, H. Niemann, S. Rieck, A. Scheuer, and E. Schukat-Talamazzini, (1993) “A Spoken Dialogue System for German Intercity Train Timetable Inquiries,” *Proc. Eurospeech-93*, pp. 1871–1874.

G. Flammia, J. Glass, M. Phillips, J. Polifroni, S. Seneff, and V. Zue, (1994) “Porting the Bilingual Voyager System to Italian,” *Proc. ICSLP-94*, pp. 911–914.

J. Glass, D. Goodine, M. Phillips, S. Sakai, S. Seneff, and V. Zue, (1993) “A Bilingual Voyager System,” *Proc. Eurospeech-93*, pp. 2063–2066.

J. Glass, J. Polifroni, and S. Seneff (1994) “Multilingual Language Generation Across Multiple Domains,” *Proc. ICSLP-94*, pp. 983–986.

D. Goddeau, E. Brill, J. Glass, C. Pao, M. Phillips, J. Polifroni, S. Seneff, and V. Zue, (1994) “Galaxy: A Human-Language Interface to On-Line Travel Information,” *Proc. ICSLP-94*, pp. 707–710.

T. Hazen and V. Zue (1994) “Recent Improvements in an Approach to Segment-Based Automatic Language Identification,” *Proc. ICSLP-94*, pp. 1883–1886.

L. Hetherington, M. Phillips, J. Glass, and V. Zue, (1993) “A^* Word Network Search for Continuous Speech Recognition,” *Proc. Eurospeech-93*, pp. 2121–2124.

W. Hutchins and H. Somers (1992) *An Introduction to Machine Translation,* Academic Press.

K. Itou, S. Hayamizu, H. Tanaka (1992) “Continuous Speech Recognition by Context-Dependent Phonetic HMM and an Efficient Algorithm for Finding N-best Sentence Hypotheses,” *Proc. ICASSP-92*, pp. 21–24.

F. Kubala, C. Barry, M. Bates, R. Bobrow, P. Fung, R. Ingria, J. Makhoul, L. Nguyen, R. Schwartz, and D. Stallard (1992) “BBN BYBLOS and HARC February 1992 Atis Benchmark Results”, *Proc. DARPA Speech and Natural Language Workshop*, pp. 72–77.

T. Morimoto, T. Takezawa, F. Yato, S. Sagayama, T. Tashiro, M. Nagata, and A. Kurematsu (1993) "ATR's Speech Translation System: ASURA," *Proc. Eurospeech-93*, pp. 1291–1294.

M. Oerder, and H. Aust, (1994), "A Realtime Prototype of an Automatic Inquiry System," *Proc. ICSLP-94*, pp. 703-706.

D. Pallett, J. Fiscus, W. Fisher, J. Garofolo, B. Lund, and M. Pryzbocki (1994) "1993 Benchmark Tests for the ARPA Spoken Language Program," *Proc. DARPA Speech and Natural Language Workshop*, pp. 49–74.

J. Peckham (1991) "Speech Understanding and Dialogue over the Telephone: an Overview of the ESPRIT SUNDIAL Project," *Proc. DARPA Speech and Natural Language Workshop*, pp. 14-27.

M. Phillips, J. Glass, and V. Zue (1991) "Automatic Learning of Lexical Representations for Sub-Word Unit Based Speech Recognition Systems," *Proc. Eurospeech-91*, pp. 577–580.

J. Polifroni, S. Seneff, and V. Zue (1991) "Collection of Spontaneous Speech Data for the ATIS Domain and Comparative Analyses of Data Collected at MIT and TI," *Proc. DARPA Speech and Natural Language Workshop*, pp. 360–365.

D. Roe, F. Pereira, R. Sproat, M. Riley, P. Moreno, and A. Macarron (1991) "Toward a Spoken Language Translator for Restricted-domain Context-free Languages," *Proc. Eurospeech-91*, pp. 1063–1066.

S. Sakai and M. Phillips (1993) "J-SUMMIT: A Japanese Segment-Based Speech Recognition System," *Proc. Eurospeech-93*, pp. 2151–2154.

S. Seneff, J. Glass, D. Goddeau, D. Goodine, L. Hirschman, H. Leung, M. Phillips, J. Polifroni, and V. Zue (1991) "Development and Preliminary Evaluation of the MIT ATIS System," *Proc. DARPA Speech and Natural Language Workshop*, pp. 88–93.

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

W. Wahlster (1993) "VERBMOBIL, Translation of Face-to-Face Dialogs," *Proc. Eurospeech-93*, pp. 29–38.

A. Waibel, A. Jain, A. McNair, H. Saito, A. Hauptmann, J. Tebelskis, (1991) "JANUS: A

Speech-to-Speech Translation System Using Connectionist and Symbolic Processing Strategies," *Proc. ICASSP-91*, pp. 793-796.

V. Zue, J. Glass, M. Phillips, and S. Seneff (1989a) "The MIT SUMMIT Speech Recognition System: A Progress Report," *Proc. DARPA Speech and Natural Language Workshop*, pp. 51–59.

V. Zue, J. Glass, M. Phillips, and S. Seneff (1989b) "The MIT VOYAGER Speech Understanding System: A Progress Report," *Proc. DARPA Speech and Natural Language Workshop*, pp. 179–189.

V. Zue, N. Daly, J. Glass, D. Goodine, H. Leung, M. Phillips, J. Polifroni, S. Seneff and M. Soclof (1989c) "The Collection and Preliminary Analysis of a Spontaneous Speech Database," *Proc. DARPA Speech and Natural Language Workshop*, pp. 126–134.

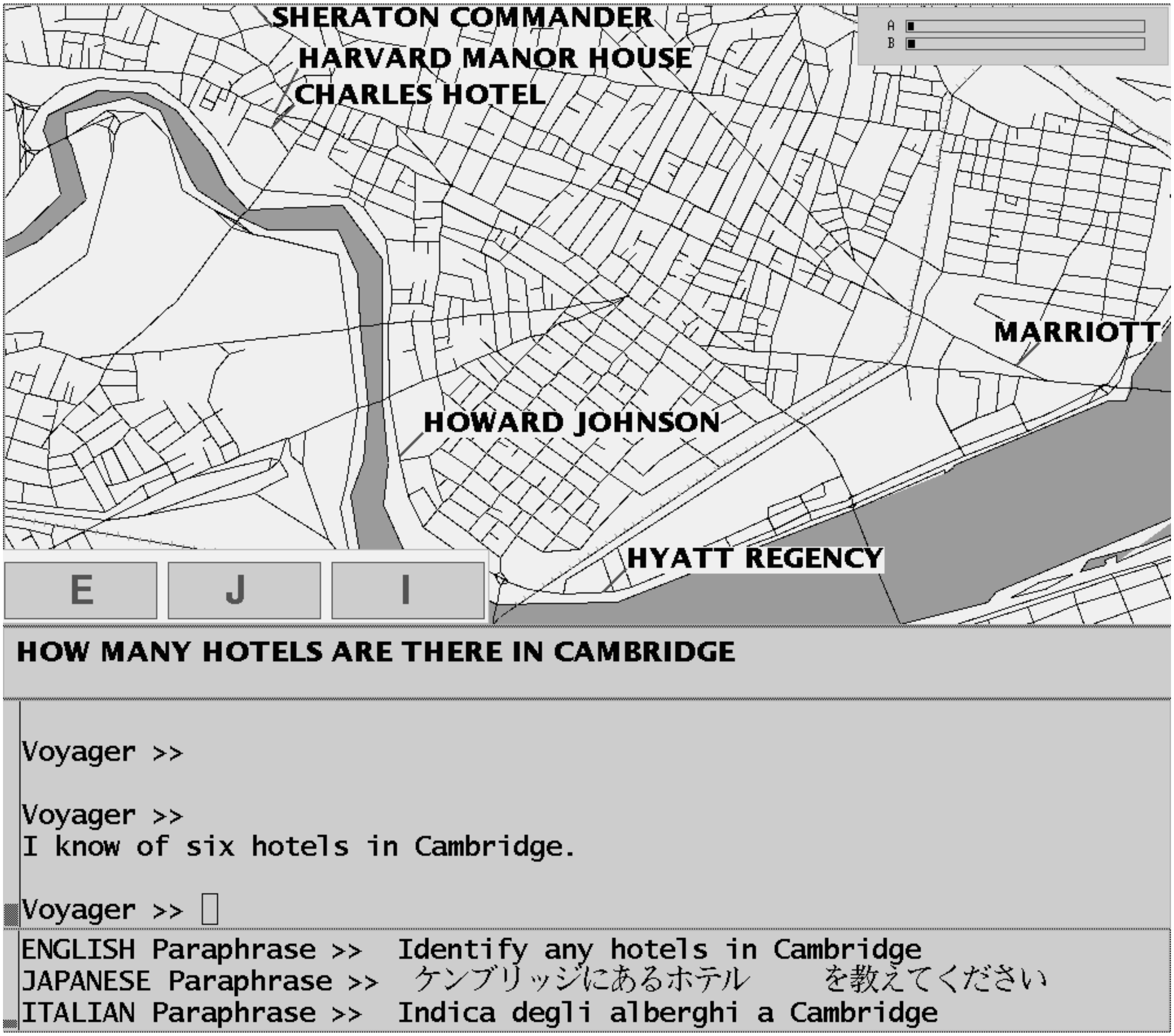
V. Zue, J. Glass, D. Goodine, H. Leung, M. Phillips, J. Polifroni, and S. Seneff (1989d) "Preliminary Evaluation of the VOYAGER Spoken Language System," *Proc. DARPA Speech and Natural Language Workshop*, pp. 160–167.

V. Zue, S. Seneff, and J. Glass (1990) "Speech database development at MIT: TIMIT and beyond," *Speech Communication,* Vol. 9, No. 4, pp. 351–356.

V. Zue, J. Glass, D. Goddeau, D. Goodine, H. Leung, M. McCandless, M. Phillips, J. Polifroni, S. Seneff, and D. Whitney (1990) "Recent Progress on the MIT VOYAGER Spoken Language System," *Proc. ICSLP-90*, pp. 1317–1320.

V. Zue, J. Glass, D. Goddeau, D. Goodine, L. Hirschman, M. Phillips, J. Polifroni, and S. Seneff (1992) "The MIT ATIS System: February 1992 Progress Report," *Proc. DARPA Speech and Natural Language Workshop*, pp. 84–88.

V. Zue, J. Glass, D. Goddeau, D. Goodine, C. Pao, M. Phillips, J. Polifroni, and S. Seneff (1993) "PEGASUS: A Spoken Dialogue Interface for One-line Air Travel Planning," *Proc. International Symposium of Spoken Dialogue.*



```
HOW MANY HOTELS ARE THERE IN CAMBRIDGE

Voyager >>

Voyager >>
I know of six hotels in Cambridge.

Voyager >>
ENGLISH Paraphrase >>  Identify any hotels in Cambridge
JAPANESE Paraphrase >>  ケンブリッジにあるホテル　　を教えてください
ITALIAN Paraphrase >>  Indica degli alberghi a Cambridge
```

Figure 1: The multilingual VOYAGER spoken-language system.
This figure shows a layout of the VOYAGER system as seen by a user. The top display contains a street map of an area of Cambridge, Massachusetts, USA. The lower left corner of the map contain the multilingual record buttons. The upper right corner contains a sound level meter. The lower windows contain the speech recognition output, the system text response, and the multilingual paraphrasing of the input query, respectively. Objects relevant to the response are highlighted in the map display.

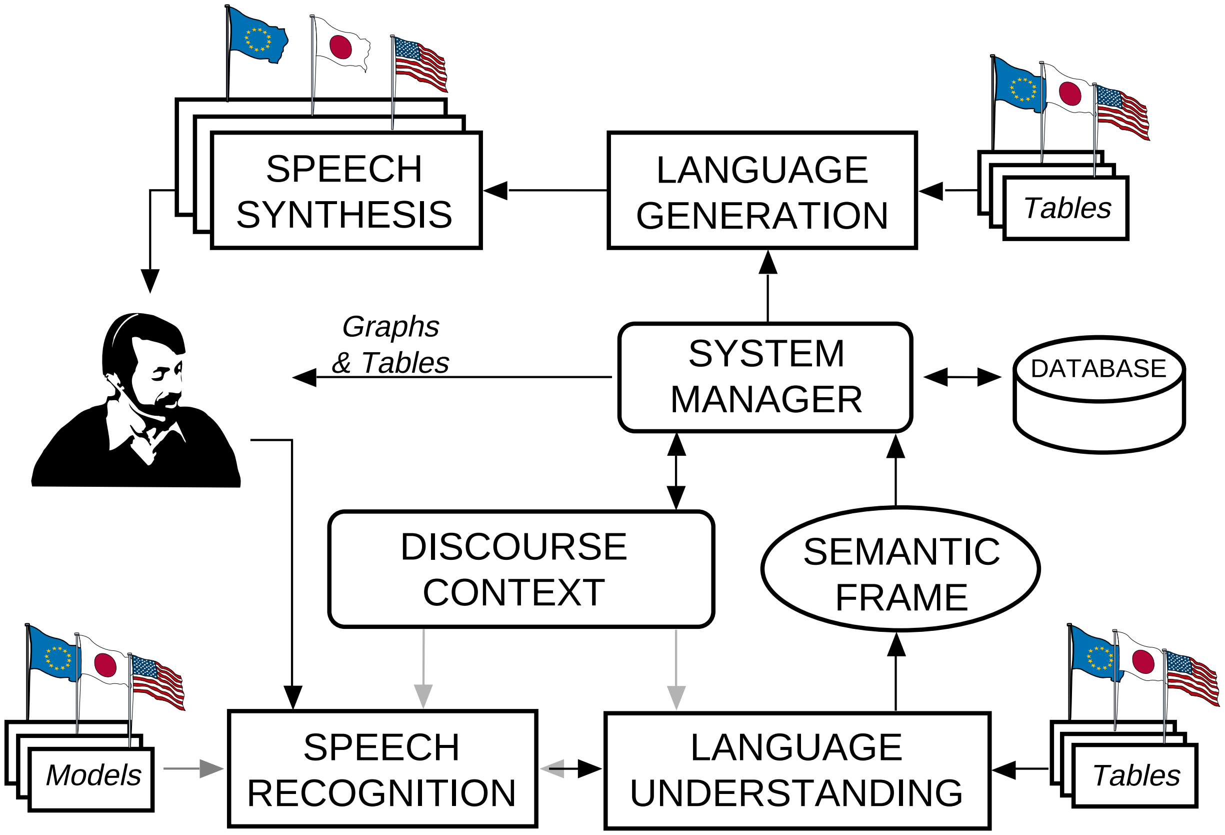


Figure 2: Schematic of prototypical MIT spoken-language system.

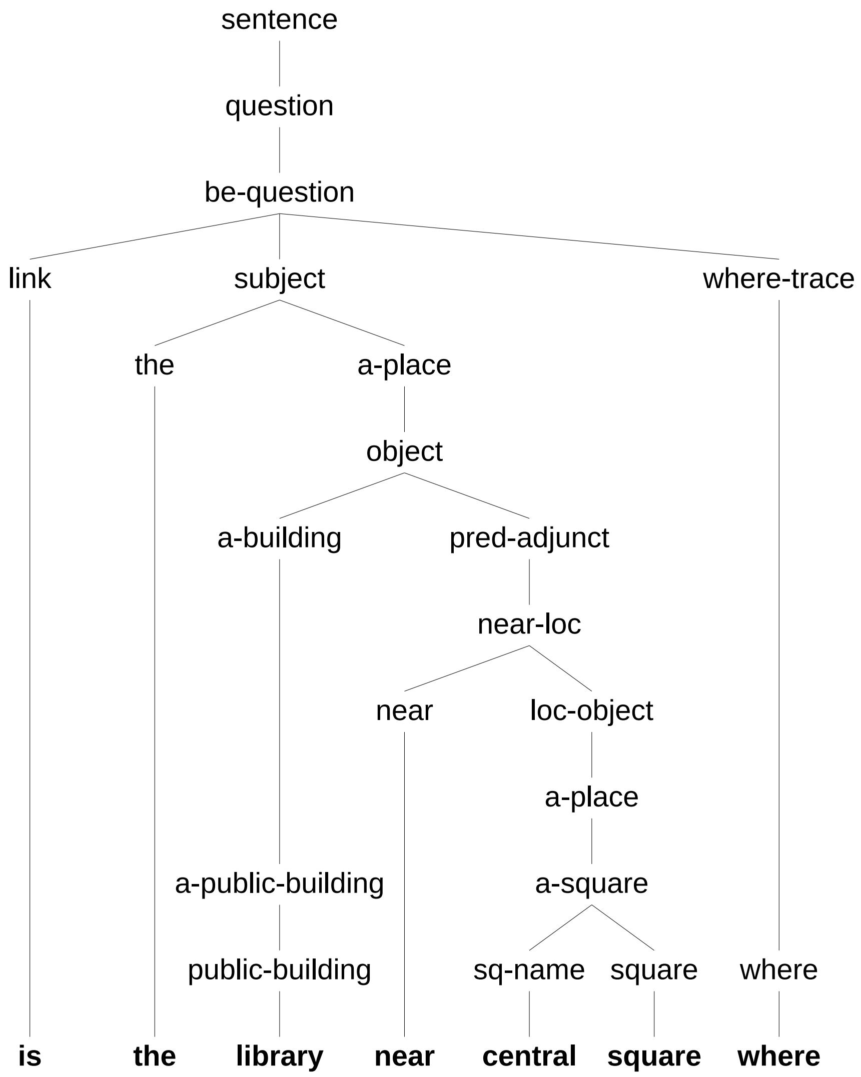


Figure 3: Parse tree for the sentence, “Where is the library near Central Square?”

```
INPUT: WHERE IS THE LIBRARY NEAR CENTRAL SQUARE
FRAME:
       Clause: LOCATE
            Topic: BUILDING
                Quantifier: DEF
                Name: library
                Predicate: NEAR
                        Topic: SQUARE
                            Name: Central
```

Figure 4: Semantic frame for the sentence, “Where is the library near Central Square?

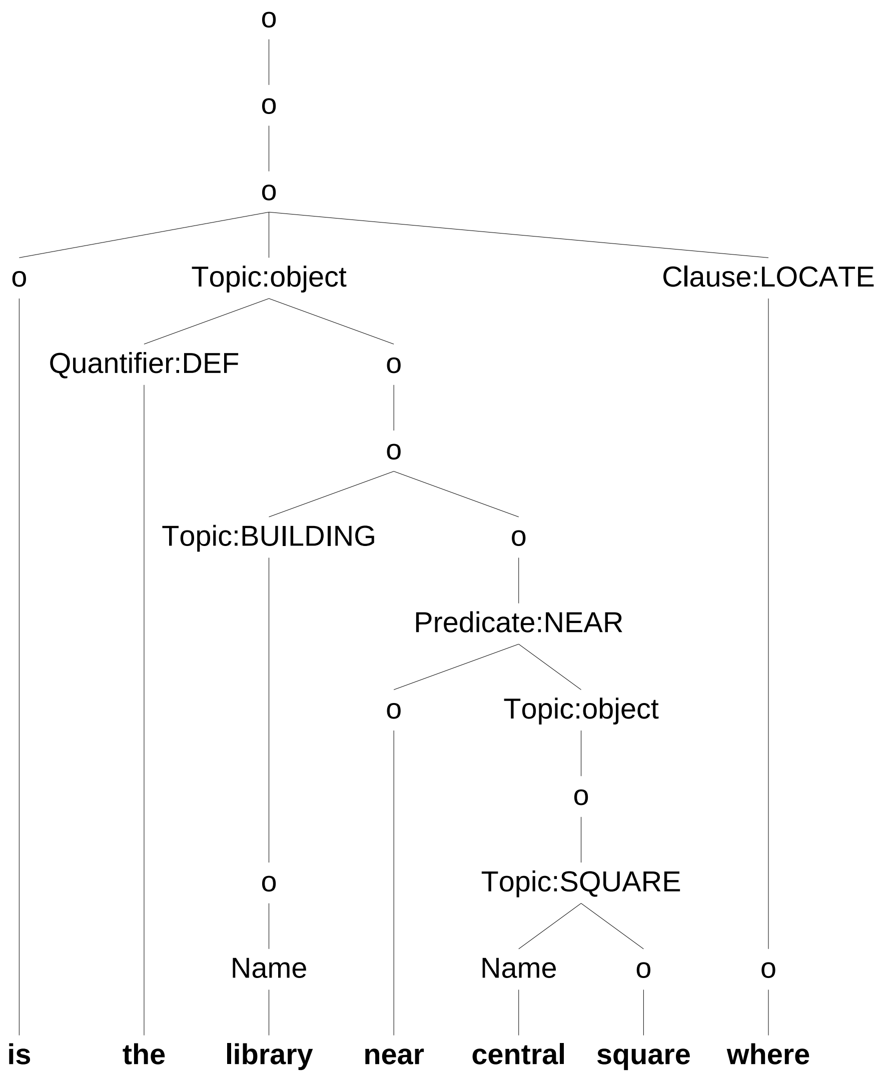


Figure 5: Semantic tree for the sentence, "Where is the library near Central Square?"

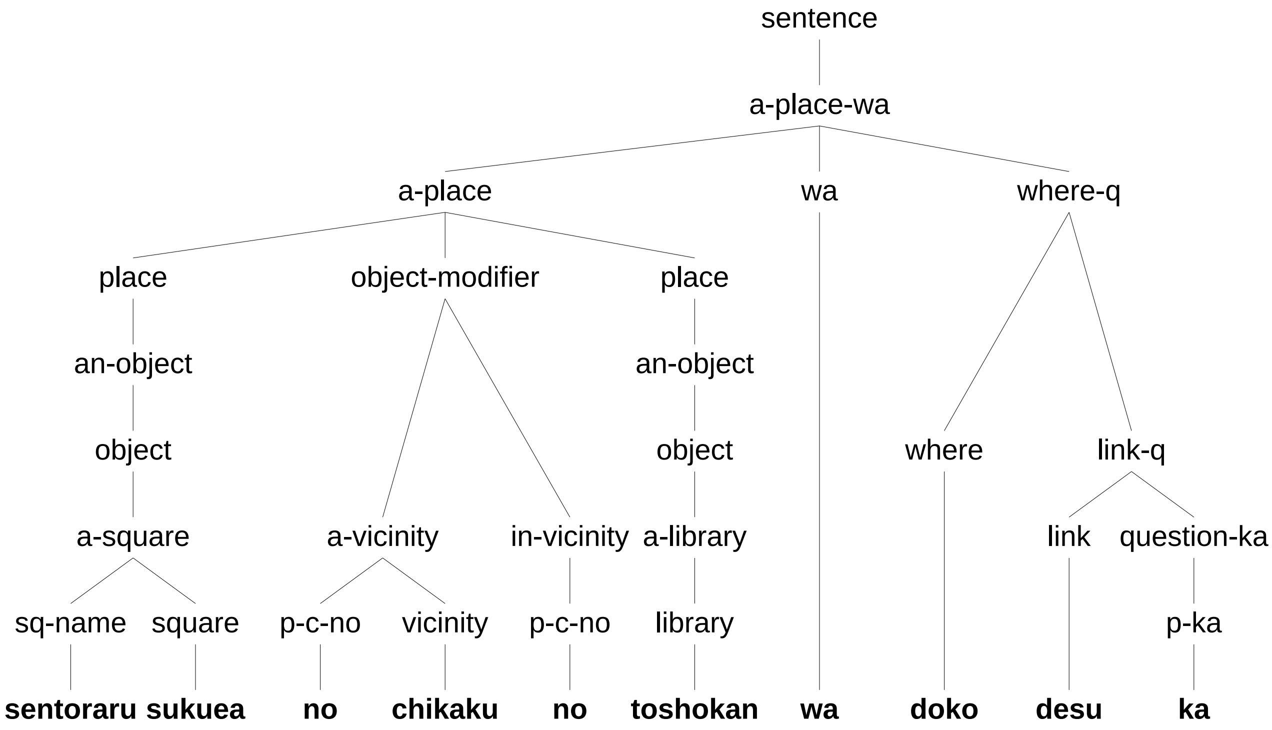


Figure 6: Parse tree for the Japanese sentence, "Sentoraru sukuea no chikaku no toshokan wa doko desu ka?"

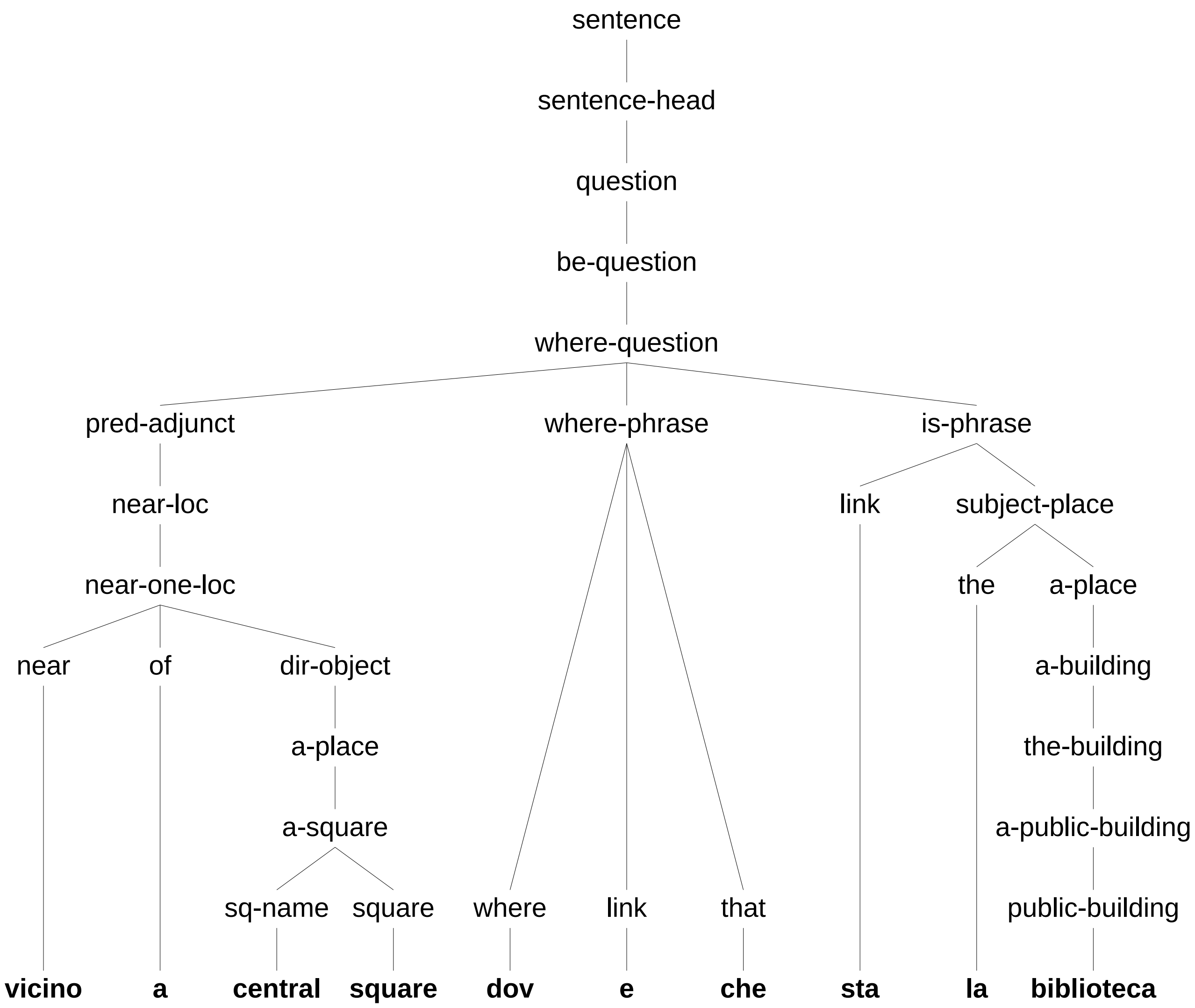


Figure 7: Parse tree for the Italian sentence, "Vicino a Central Square, dove sta la biblioteca?"

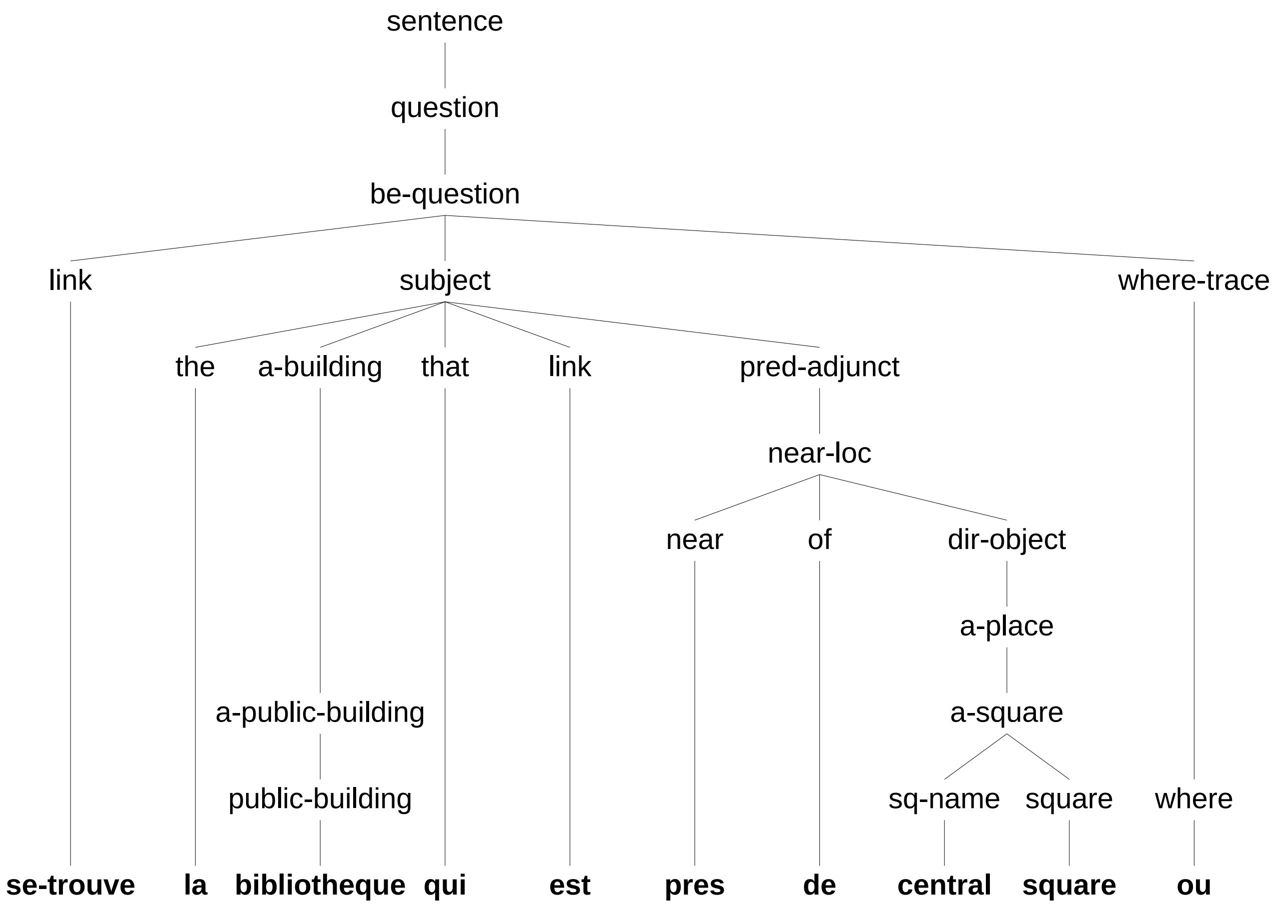


Figure 8: Parse tree for the French sentence, “Où se trouve la bibliotheque qui est près de Central Square?”

English Lexicon:		
V	V	“Verb” THIRD “es” ROOT “e” ING “ing”...
N	N	“Noun” PL “s”
be	AUX	“be” ROOT “be” THIRD “is” ING “being”...
do	AUX	“do” THIRD “does”... MODE “root”...
indef	Q	“a” PL “any”
which	TRACE	“what”
royal_east	P	“Royal East” Q “def”
serve	V	“serv”
on_street	PREP	“on”
French Lexicon:		
N	N	“Noun” PL “s” F “e” FPL “es”
V6	V	“Verb” ROOT “vir” THIRD “t” FPL “vons”...
be	AUX	“etre” ... FPL “sommes”...
which	TRACE	“quel” F “quelle” MPL “quels” FPL “quelles”
royal_east	P	“Restaurant Royal East” Q “def”
serve	V6	“ser”
on_street	PREP	“dans”

Table 1: Selected entries from the generation lexicon for English and French. Each entry consists of a name, a part of speech, a default text realization, and optional keyword value pairs indicating either alternative realizations (e.g., the feminine realization of the word “which” in French is “quelle”), or lexical properties of the entry (e.g., the preferred realization of the word “royal_east” is with a definite article).

existential	(:AUX be) there :TOPIC .
wh_query (English)	:TRACE (:AUX be) (:TOPIC it) :PRED :PREP ?
wh_query (French)	:PREP :TRACE (:PRED be) :TOPIC ?
topic	:QUANTIFIER :NOUN_PHRASE
street	:TOPIC :STREET_TYPE
serve	:PREDICATE :TOPIC
np-on_street	:NOUN_PHRASE :PREDICATE :TOPIC

Table 2: Selected generation message templates. Each entry consists of a name (e.g., existential) and a sequence of words and/or keywords. Words are represented in lowercase text (e.g., "there"). Their actual realization is determined via the lexicon. Keywords are represented by uppercase text (e.g., :TOPIC). Their values are determined via recursive evaluation of keywords in the semantic frame. Note that default values are available in the event no keyword value is available (e.g., (:AUX be) uses the verb "be" as a default if there is no value for the :AUX keyword in the semantic frame).

		Left	Right
Word ID	Pronunciation	Category	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

Table 3: Example lexical entries for the Japanese recognizer. Each lexical entry consists of a word ID, a pronunciation, and left and right morphological categories.