Original Investigations JAMIA

Research Paper ■

A Continuous-speech Interface to a Decision Support System: I. Techniques to Accommodate for Misrecognized Input

SMADAR SHIFFMAN, MS, WILLIAM M. DETMER, MD, CHRISTOPHER D. LANE, LAWRENCE M. FAGAN, MD, PHD

Abstract **Objective**: Develop a continuous-speech interface that allows flexible input of clinical findings into a medical diagnostic application.

Design: The authors' program allows users to enter clinical findings using their own vernacular. It displays from the diagnostic program's controlled vocabulary a list of terms that most closely matches the input, and allows the user to select the single best term. The interface program includes two components: a speech-recognition component that converts utterances into text strings, and a language-processing component that matches recognized text strings with controlled-vocabulary terms. The speech-recognition component is composed of commercially available speech-recognition hardware and software, and developer-created grammars, which specify the language to be recognized. The language-processing component is composed of a translator, which extracts a canonical form from both recognized text strings and controlled-vocabulary terms, and a matcher, which measures the similarity between the two canonical forms.

Results: The authors discovered that grammars constructed by a physician, who could anticipate how users might speak findings, supported speech recognition better than did grammars constructed programmatically from the controlled vocabulary. However, this programmatic method of grammar construction was more time efficient and better supported long-term maintenance of the grammars. The authors also found that language-processing techniques recovered some of the information lost due to speech misrecognition, but were dependent on the completeness of supporting synonym dictionaries.

Conclusions: The authors' program demonstrated the feasibility of using continuous speech to enter findings into a medical application. However, improvements in speech-recognition technology and language-processing techniques are needed before natural continuous speech becomes an acceptable input modality for clinical applications.

■ **J Am Med Informatics Assoc.** 1995;2:36–45.

Affiliation of the authors: Section on Medical Informatics, Stanford University, Stanford, CA.

Presented in part at the Sixteenth Annual Symposium on Computer Applications in Medical Care, Baltimore, Maryland, 1992.

Supported by the National Library of Medicine under grants LM-04864 and LM-07033. Computer facilities were provided by the SUMEX-AIM project (LM-05208), by CAMIS Resource (LM-05305), and through an equipment load from Speech Systems, Inc.

Correspondence and reprints: Smadar Shiffman, MS, Section on Medical Informatics, Medical School Office Building X215, Stanford University, Stanford, CA 94305-5479. e-mail: shiffman@camis.stanford.edu

Received for publication: 5/31/94; accepted for publication: 9/07/94.

Many health professionals resist using medical decision support applications because of the effort they must expend to learn how to operate these applications. Barriers to use include input with a keyboard or a pointing device, and for applications that use controlled-vocabulary terms, the necessity to learn the vocabulary. A previous study suggested that health care providers might use medical applications more often if speech, rather than conventional input techniques, were the interface modality.[1] Ideally, users could speak continuous natural-language sentences and that input would be translated into the controlled vocabulary used by the decision support application.

Commercially available speech-recognition systems require two complementary specifications of the language to be recognized: a *grammar*, or rules for sentence construction, and a *vocabulary*, or a list of allowable words. From these specifications speech-recognition systems produce an inventory of acoustic patterns for all sentences that the systems can recognize, and label each pattern with a textual representation of the pattern. To recognize speech, the systems generate an acoustic-pattern representation of an utterance, and compare the pattern with the acoustic patterns in the inventory. The systems designate the textual label of the closest pattern as the best guess of what was spoken. The variety of sounds that are produced when different people speak the same sentence or when the same person speaks a sentence at different times introduces uncertainty into the speech-recognition process, which results in a risk of misrecognition.

Medically oriented speech applications have reduced the risk of misrecognition by requiring short utterances (individual words or a few words in sequence) and by exploiting domain areas where the expected language is limited, structured, and well defined.[2] The imposition of similar constraints on a speech interface for the entry of clinical findings would most likely cause the interface to be unusable, both because the domains of typical decision support tools are broad and because the language that physicians use to define clinical findings is broad. We experimented with a speech-recognition system that supported continuous speech with large vocabularies, so that the language physicians could use to enter findings would be minimally restricted.

We designed grammars that specified the language we expected physicians to use to describe findings. We anticipated that physicians would say sentences that were not included in the language specification and that spoken utterances would be partially misrecognized. We accommodated for misrecognition by using a matching procedure to match misrecognized input utterances to controlled-vocabulary terms. In this article we describe the techniques we used to build the speech interface and the lessons we learned from that experience. A companion article in this issue describes a study that we performed to evaluate the speech interface.[3]

Background

Recent advances in speech-recognition technology made it feasible to use continuous speech as an input modality for medical applications.[4,5] We developed a continuous-speech interface that allowed the user to enter *findings*, or clinical observations about a patient, into a medical diagnostic program. We built this speech interface for use with the knowledge base of Quick Medical Reference (QMR),*[6,7] a well-known medical diagnostic program. We selected the QMR knowledge base because it uses a controlled vocabulary, and we believed that a continuous-speech interface could facilitate flexible input of clinical findings from a controlled vocabulary. The focus of our effort was to develop a speech interface for an existing diagnostic system while using available speech-recognition technology, rather than to research speech-recognition algorithms.

QMR Knowledge Base

The QMR program provides access to a database of internal-medicine diseases. It was derived from an earlier system, INTERNIST-1,[6,7] and includes 4,000 history, physical examination, and laboratory test findings; 600 diseases; and links that define causal, temporal, and logical interrelationships among diseases. The findings in the QMR knowledge base are expressed as compound-noun phrases. Some of these phrases are ungrammatical or awkward (e.g., *live fine nodule*); others may not be familiar to users (e.g., *abdomen flank bulging bilaterally*). One of the ways physicians can interact with the standard interface to QMR is by entering a list of findings; the system then provides differential diagnoses that are ranked by their likelihood. The QMR program provides a typing-based search tool, which physicians can use to enter a set of keywords. The program retrieves terms

*Quick Medical Reference and QMR are registered trademarks of the University of Pittsburgh. The vocabulary of QMR was derived from the findings in INTERNIST-1. We used the terms from INTERNIST-1 for our research, but we have described our research as a spoken interface to the QMR vocabulary because we believed that the terms from INTERNIST-1 were representative of the QMR vocabulary.

from the knowledge base that match the typed input. The user can then select appropriate terms from the retrieved list for entry into the diagnostic program. The QMR program addresses the expected variability of input by using a synonym dictionary as part of a search tool that accepts prefixes or acronyms of terms from the user. In contrast to the standard QMR interface, our interface used speech input and a matching procedure to match findings with terms from the QMR knowledge base.

Linguistic Methods for Processing Medical Vocabulary

Semantic analysis is a process that extracts the meanings of phrases from their forms. Linguistic methods that perform semantic analysis of textual phrases have been used successfully within applications that incorporate medical terminology. The Linguistic String Project[8] used semantic techniques in computer applications that managed narrative data. The CLARIT project[9] and the SAPHIRE information-retrieval system[10] applied semantic approaches to perform automatic indexing. The METEXA system[11] used semantic knowledge to support speech processing through a conceptual model of radiologic reports. Our approach was similar to these approaches in its reliance on the content of expressions rather than on the surface form of expressions. Our method extracted the key concepts in a phrase to form an underlying semantic representation of findings.

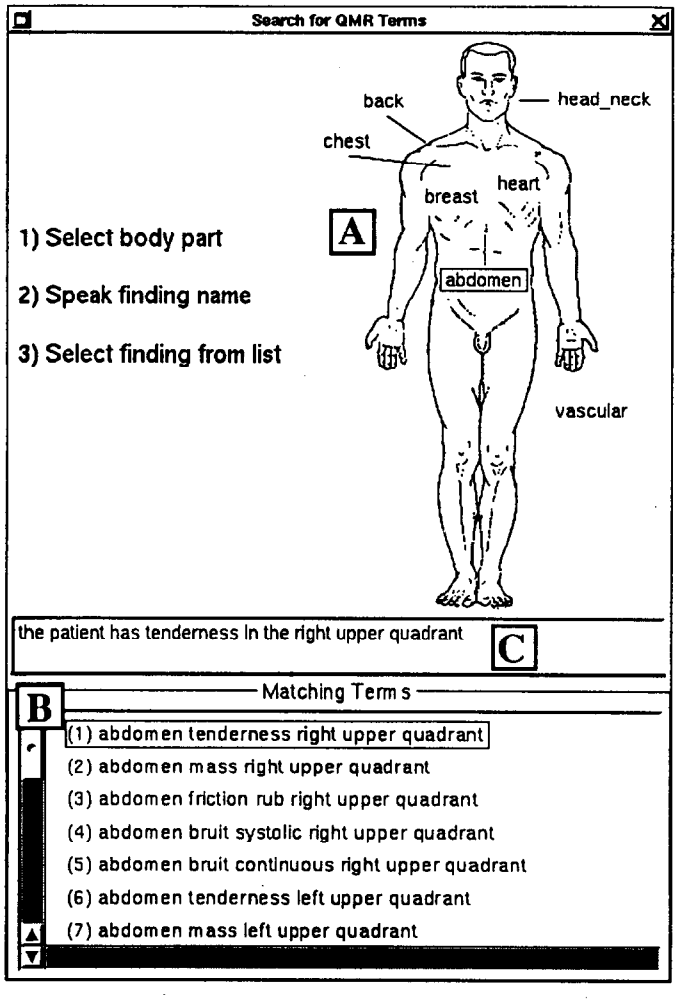

Figure 1 Computer display of the continuous-speech interface to Quick Medical Reference (QMR). To add a clinical finding to the QMR case description, the user selects a body part (A) by speaking or by using a pointing device, speaks a finding into the speech apparatus, and chooses the QMR finding (B) that best corresponds to the intended clinical finding. The middle box (C) contains the text string that is generated by the speech recognizer.

Speech-recognition Systems

Speech-recognition systems can recognize vocabularies ranging from tens to thousands of words.[12] *Speaker-dependent* systems accept input only from a specific speaker, whereas *speaker-independent* systems accept input from any speaker. *Isolated-word* systems requires that speakers pause between words or short phrases, whereas *continuous-speech* systems allow speakers to utter long sequences of words without pausing. Most speech-recognition systems in common use incorporate isolated-word technology.[2]

The usefulness of speech-recognition systems as user interfaces depends in part on the interpretation of *recognized utterances*, or textual representations of utterances that these systems produce. Research projects have used a variety of natural-language processing techniques to interpret textual strings. Simple interpretation techniques rely on template matching[13]; more complex techniques use broad linguistic knowledge including syntax, semantics, and pragmatics.[14]

We mentioned earlier in this article that continuous-speech recognition systems require target-language specifications in the form of grammars and vocabularies. Common grammar forms are phrase-structure rules[15] and trigrams (triplets of words and associated probabilities that indicate the probability that a given word follows its two precedents within a single sentence). The form of a grammar represents a compromise between conflicting trends that affect the accuracy of a speech system: as the size and complexity of a grammar increase, the grammar generates more speech patterns and therefore the system recognizes more variable input. However, the large number of possible patterns increases the probability of confusion between similar patterns, and therefore the recognition accuracy decreases. Misrecognition is certain to occur for utterances that are not represented in a grammar.

Design Considerations

Our research tasks were to develop grammars that could support the recognition of spoken sentences, and to design a method for matching recognized utterances to controlled-vocabulary terms. Variability in user language, differences in the detail included in input utterances and controlled-vocabulary terms, and imperfect performance of the speech-recognition system influenced the methods we developed. We describe our considerations below.

Variability of Input Language

Physicians may use different expressions to specify a single finding. For example, a physician describing a mass in the right upper side of the abdomen might use the sentence *I noticed a small right upper quadrant mass*, or the sentence *a lump was felt in the right upper quadrant*. The corresponding term found in QMR is *abdomen mass right upper quadrant*. The absence of standard terminology for expressing findings and the number of terms in the knowledge base require users to speculate on the terms found in the system. It has been our experience[5] that user interfaces must be tuned for each input modality. The QMR keyboard interface for entering terms is able to take advantage of abbreviations such as RUQ for *right upper quadrant* that may be awkward when given as spoken commands.

Continuous-speech recognition systems specify the language to be recognized explicitly, in the form of a grammar. Specification of the language requires system developers to predict what sentences physicians would use to describe findings. Although physicians could express a finding in many ways, in reality, conventions of medical language limit the ways physicians phrase findings, and therefore we believed the task of predicting physician expression of findings was feasible.

We realized that to define a grammar that would be small enough to ensure speech recognition with reasonable accuracy, we would have to select a subset of the QMR terms and partition that subset into smaller subsets, each of which would have its own target sublanguage and subgrammar.

Variability of Detail

The main task for our interface was to interpret input utterances and to find controlled-vocabulary terms from the QMR knowledge base that matched the input. We expected that input utterances would not match exactly any term in QMR, either because the input utterances were less specific than the terms in QMR or because the utterances were more specific than comparable terms in QMR. We anticipated that misrecognition would distort the information conveyed by the input utterance. We were aware that the ASCII string produced by the speech-recognition system for the input utterance might not include some of the words that were spoken, or might include words that were not spoken. Our interface needed a matching process that would recognize correspondence between input utterances and target controlled-vocabulary terms despite differences in information content.

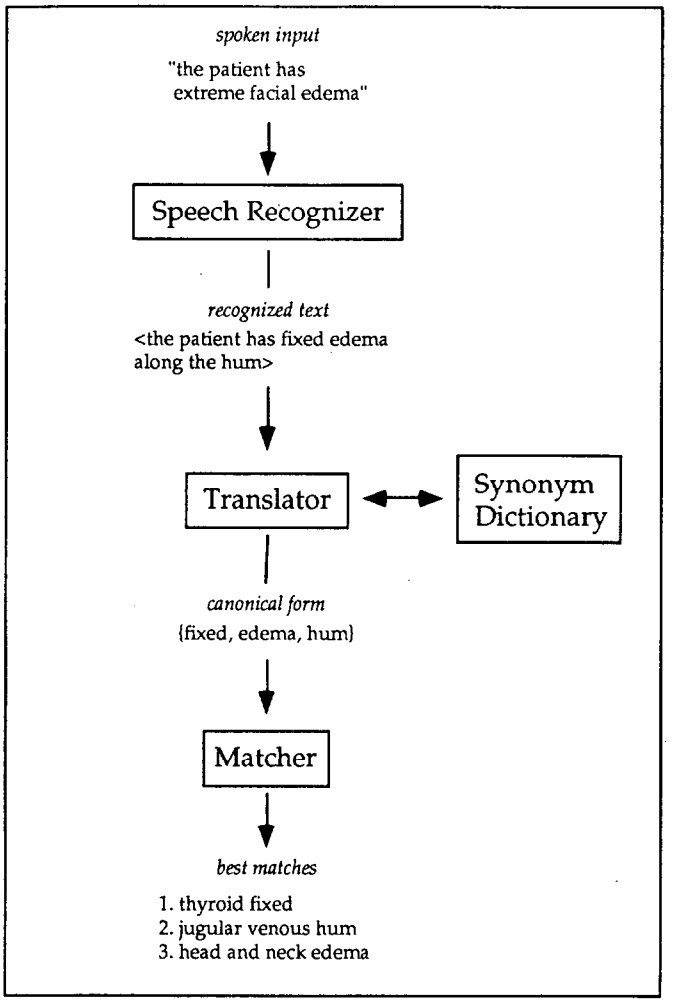

Figure 2 System architecture for the interface program: the sequential operation of system modules for analyzing input utterances and identifying corresponding Quick Medical Reference (QMR) terms. Each step is accompanied by an example output.

Representation of the Target Language in the Form of a Grammar

The construction of a natural-language grammar requires the use of linguistic knowledge to form a con-

cise formal description that would account for all sentences in the language and only these sentences. Linguists commonly compose grammars from production rules that can generate the unlimited number of sentences included in natural languages.[15] However, the use of production rules for describing natural languages often results in inaccurate specifications of these languages: individual rules might be overly powerful in that they generate sentences that are not part of the language, and the set of rules as a whole might not be powerful enough to capture all the sentences included in a natural language.

The construction of grammars for supporting speech recognition required that we find a satisfying balance between developing a large grammar that would be expressive and a small grammar that would support accurate recognition.

Methods

We used a keyword-based semantic representation for finding names and a matching process to map input utterances to terms in QMR. We assumed that even if utterances were partially misrecognized, enough of the semantic content would remain to allow the system to match the utterances to controlled-vocabulary terms that were similar in meaning. To simplify our task, we assumed that physicians would input findings in simple sentences in affirmative form—for example, we assumed physicians would say *the patient had a mass in the abdomen* and would not say *the patient reported pain in the abdomen and the abdomen was tender upon palpation.*

System Architecture

The interface program we built included the following three modules:

1. A commercially available speech-recognition system that produced ASCII strings from speech signals.
2. A translator that generated keyword-based canonical forms from recognized strings and from terms in QMR. The translator induced a procedure for looking up words in a synonym dictionary.
3. A matcher that compared canonical forms of input utterances and canonical forms of terms in QMR, and produced a score for each match.

The three modules supported the following interaction cycle for entering a finding to a case description (Fig. 1). First, the physician selected a body part and spoke a finding into a head-mounted microphone, which transferred the acoustic signal to the speech-recognition system. Then the speech-recognition system converted the utterance into an ASCII string. Next, the translator extracted the essence of the ASCII string into a keyword-based canonical form. Then the matcher compared the canonical form that originated from the input with similar precomposed forms of terms in QMR. At this point, the program displayed the result of the matching as a rank-ordered list of terms from QMR. Finally, the user selected a finding from several offered to add to the case description. When the program could not find an appropriate matching term from QMR for an input utterance because the utterance had been misrecognized completely, the user could repeat the utterance or could edit the string returned by the speech-recognition system and reenter the edited string. Figure 2 demonstrates the system architecture that we used for our interface.

Translation of Finding Names into Canonical Forms

The interface program captured the key concepts from a recognized utterance in a canonical form by con-

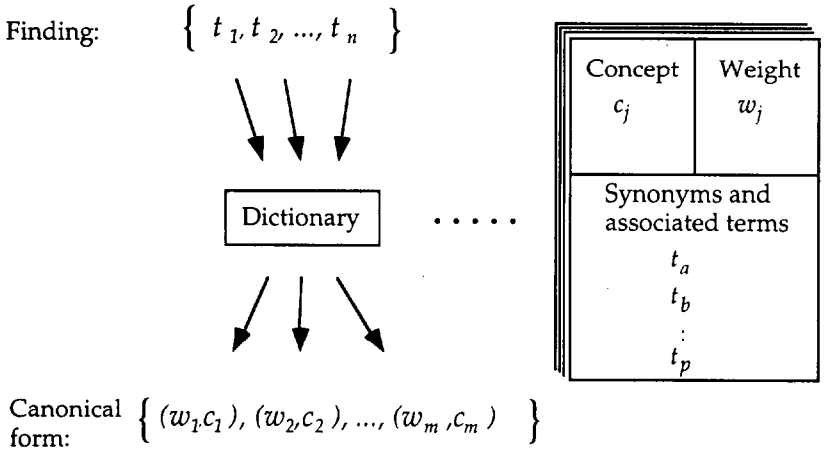

Figure 3 Translation of a recognized utterance into a canonical form through a lookup in a synonym dictionary. The structure of the dictionary is shown on the right: t_i designates tokens of the recognized string, c_j designates concepts that represent synonyms and related terms, and w_j designates weights associated with concepts.

Figure 4 Typical grammar rules from a grammar that describe possible utterances for physical examination findings related to neck bruits. Words in uppercase letters designate classes of words that represent a single concept. Words in lowercase letters separated by the *or* sign (|) designate members of a class of synonyms or terms related to the concept in uppercase letters. Words in parentheses designate optional elements.

BRUIT_PHRASE -> (INTENSITY) (CYCLE) (REGION) bruit

INTENSITY-> soft | loud | faint

CYCLE -> systolic | holosystolic | pansystolic | diastolic | continuous

REGION -> neck | carotid | thyroid

structing a set of related *keywords* or key concepts. For example, the program represented the term *the liver is enlarged on examination* by the canonical form {*hepar, enlargement*}, where *hepar* designated the concept *liver* and *enlargement* designated the concept *enlarged*. The translator produced canonical representations for terms by identifying the set of meaningful words in each recognized string through a lookup in a dictionary that included keywords and their synonyms. The emphasis on key concepts rather than on words allowed the program to identify sentences of similar meaning independent of their forms.

The dictionary included a list of concepts, each of which represented a set of synonyms. For example, the concept *uterine* represented the class of terms *uterine, uterus, endometrial,* and *womb*. We generated the dictionary by extracting a set of concepts from the dictionary used in QMR and by adding synonyms to the list of concepts through manual editing. The representative concept for a list of synonyms was chosen arbitrarily. We observed that infrequent concepts were more informative in that they designated matching terms with greater certainty. We assigned to each concept within the dictionary a weight that was inversely related to the frequency of the concept in the QMR knowledge base. The weighting formula emphasized the importance of specific terms by assigning larger weights to these terms.

We computed weights for concepts in our set of terms from QMR according to the formula

$$w_i = -\log \frac{c_i}{n}$$

where w_i is the weight of concept i, c_i is the number of occurrences of words and word combinations that represent concept i in the knowledge base, and n is the total number of words or word combinations in the knowledge base. This weighting formula is commonly used in information-retrieval methods,[16] but, to our knowledge, it is not commonly used in speech-based applications.

For every finding processed, the translator identified words that appeared in the synonym dictionary (either in isolation or as word combinations) as keywords and included the representative of the synonym class to which the keywords belonged in the canonical form. The translator ignored words that did not appear in the synonym dictionary. The canonical form comprised pairs of keywords and their associated weights. Figure 3 illustrates the translation process.

Matching Canonical Forms

When a finding was spoken into the microphone, the matcher computed a score for every term in QMR based on the distance between the newly generated canonical form for the input (t for test canonical form) and the precomputed canonical forms for target terms (r for reference canonical forms). The distance measure was based on the assumption that similar terms included the same concepts, and that dissimilar terms included different concepts. The distance formula was a function of the specificity of concepts, as manifested by concept weights:

$$D(t, r) = \sum_{i=1}^{l} w_i - \sum_{j=1}^{m} w_j - \sum_{k=1}^{n} w_k$$

For this formula, l is the number of concepts that appear in both canonical forms, m is the number of concepts that appear only in the canonical form for the input utterance, and n is the number of concepts that appear only in the canonical form for the controlled-vocabulary term. The scoring formula assigned a reward or a penalty to the score, depending on whether concepts in the input canonical form were included in the canonical form for the target term or were excluded from the canonical form. Thus, the formula produced higher scores when target terms included highly specific concepts and when the canonical form of the recognized text matched closely to the canonical form of the term in QMR.

System Configuration

The speech interface displayed on an NeXT workstation (NeXT Computer Inc., Redwood City, CA) that was connected to a Speech Systems, Inc. (Tarzana, CA) (SSI) DS200 speech-recognition system. The SSI system used a Sun SPARCstation to decode

A

```
S --> (DC_BEGIN) MASS_M
S --> (DC_BEGIN) BREAST_M (DC_MID) MASS_M
S --> (DC_BEGIN) MASS_M (DC_MID) BREAST_M
S --> (DC_BEGIN) BILATERAL_M (DC_MID) BREAST_M
S --> (DC_BEGIN) MASS_M (DC_MID) BILATERAL_M
S --> (DC_BEGIN) BILATERAL_M (DC_MID) MASS_M (DC_MID) BREAST_M

BREAST_M > breast
MASS_M > mass | nodule | lump
BILATERAL_M > bilateral | both sides
```

B

```
DC_BEGIN --> i {found | noticed | notice | saw | see} ({a | some | many | that the})
DC_BEGIN --> {the patient | he | she} {has} {a | some | many}
DC_BEGIN --> the | it
DC_MID --> is
DC_MID --> PREP {the | her | his | the patient's | a}
PREP == in | on | about | by | with | at | under | over | of
```

Figure 5 Sample grammar rules that were derived programmatically from the canonical form {*breast, mass, bilateral*}, which represents the QMR term *breast mass bilateral*. The rules in group A were generated programmatically. The rules in group B, which added nonclinical terms that provide flexibility in sentence structure, were created once manually and then added to each of the programmatically generated grammars. All terms marked with the suffix _M are nonterminal symbols. The symbols DC_BEGIN represent classes of words that could appear at the beginning of an utterance and DC_MID the classes of words that could appear in the middle of an utterance.

utterances. The SSI system included a vocabulary of more than 38,000 words (including root forms and inflections), from which we used about 1,000 words. It recognized continuous speech and was speaker-independent. The SSI system required a specification of a vocabulary and a grammar for each set of sentences that it could recognize.

Construction of Grammars

The SSI system required that the set of recognizable input sentences be specified as grammars of formal languages. A formal language is a set of finite-length strings formed from a finite vocabulary.[15] Grammars of formal languages are specified in terms of syntactic categories, such as ⟨NOUN_PHRASE⟩, terminal symbols from the language vocabulary, and production rules that specify the relation between syntactic categories and terminal symbols. Typically, production rules are designated by a set of symbols: a syntactic category followed by an arrow and a list of syntactic categories and/or terminal symbols. This representation signifies that the syntactic category may be replaced by all that follows the arrow. Production rules may be applied in sequence to generate or to parse natural-language expressions. Figure 4 shows a list of typical rules we included in grammars that we constructed for the SSI speech-recognition system.

We used two approaches to generate grammars for the speech-recognition system. First, one of us (WMD) tailored grammars manually for the QMR domain by reviewing the QMR terminology and estimating subjectively how health care providers might phrase controlled-vocabulary terms in natural language. For example, sentences that users could say to describe a bruit in the neck area, such as *soft diastolic carotid bruit, neck bruit,* or *loud continuous bruit,* were anticipated and the rules in Figure 4 were extracted, which could generate these sentences. Then rules were extracted by generalizing sets of words into semantic categories, and determining possible orderings for these categories.

Second, we generated grammars programmatically by deriving phrase-structure rules from the target terms in QMR. We considered each term in QMR to be a target finding that the user could express using a variety of words and word orders. We generated grammars that captured the expected variety of expressions by applying a sequence of transformations to the canonical form of each target term in QMR:

1. We generated the *power set*, or the set of all subsets, of keywords for the canonical form to account for use of only some of the relevant keywords in the specification of a finding. For example, the power set for the canonical form {*breast, mass, bilateral*} included {*breast, mass, bilateral*}, {*breast, mass*}, {*mass, bilateral*}, {*breast, bilateral*}, {*breast*}, {*mass*}, and {*bilateral*}. We expected that physicians would not use every keyword in specifying a finding when their notion of the particular finding was more general than the notion included in the target term. However, the resulting power set included subsets that accounted for sentences that physicians were not likely to say to describe a medical finding. For example, it was not likely that a physician would describe a finding in the breast by saying a sentence such as *the patient had a bilateral breast*, which would include only the keywords {*breast, bilateral*}.

2. We permuted each subset of keywords to account for variations in word order. For example, the permutations that resulted from {*mass, bilateral*} were {*mass, bilateral*} and {*bilateral, mass*}. These permutations accounted for sentences such as *there was a mass bilaterally* and *I noticed a bilateral mass.*
3. We substituted synonyms from the synonym dictionary for the original keywords to account for variations in the choice of words for a particular finding. For example, the substitutions *mass* → *nodule* or *bilateral* → *both sides* accounted for the use of the words *nodule* and *both sides* to describe the finding of a mass in the breast.
4. We inserted words that could be used in a specification of a finding before, between, or after keywords. For example, the insertion of the phrase *the patient had a* before the keyword *mass* and the phrase *in the* before the keyword *breast* accounted for the sentence *the patient had a mass in the breast.* Figure 5 illustrates part of the grammar that was generated by applying the transformations to the canonical form {*breast, mass, bilateral*}.

Both manually generated grammars and programmatically generated grammars were used by the SSI system to enumerate possible sentences. The SSI system generated the possible sentences by recursively expanding syntactic categories according to rules that applied to these syntactic categories. Although only a portion of the expanded sentences was stored in the computer memory simultaneously, the computation necessary to select that portion and to match the input utterance against that portion was intensive.

To simplify the grammars, we excluded the generation and interpretation of morphologic variations of words from the scope of our grammar-generation procedure. Nevertheless, the grammars that we generated included many possible sentences, some of which physicians would never say, such as *i noticed a both side breast.*

To ensure that the grammars would not be overly large, consequently yielding poor recognition, we selected a subset of the terms in QMR and partitioned the subset further so that each partition would not exceed 50 terms. We selected the domain of physical examination findings because this domain could be partitioned easily in the subdomains of body parts, and these subdomains were intuitive to the user.†

†We then excluded all findings that included words not in the vocabulary of the speech-recognition system. Thus, we used only about half of the physical examination findings (518) that were in the INTERNIST-1 knowledge base.

Lessons Learned

To evaluate our speech interface, we examined the extent to which users were able to enter QMR findings with speech. The details of the evaluation study and the results are included in the companion article by Detmer et al. in this issue.[3] In this section we describe the benefits of using a keyword approach, discuss unsuccessful attempts to construct the synonym dictionary programmatically, and present a comparison of grammar construction techniques and results.

Experiences with Keyword Matching

The method of matching keyword-based canonical forms allowed partially misrecognized utterances to still match with appropriate QMR terms. Synonyms facilitated this process by expanding the number of possible utterances that could match to a QMR term. In addition, the matching method ignored all keywords that were misrecognized as nonkeywords. Therefore, only a portion of the misrecognized words, namely, keywords that were misrecognized as other keywords, affected the accuracy of the matching process.

The emphasis on keywords allowed the system to identify related terms that differed in detail. For example, the input utterance *crescendo decrescendo diastolic murmur* did not match any target term exactly. However, the system encoded the recognized string for the utterance as the canonical form {*decrescendo, diastolic, murmur*} and elicited the more general target term *heart murmur present* and the more specific term *heart murmur diastolic decrescendo second left interspace,* both of which are relevant to the input utterance. The ability of the matching method to associate a variety of textual strings, some of which were grammatically incorrect, to a fixed set of controlled-vocabulary terms suggests that the method is suitable for supporting the integration of speech interfaces into applications that depend on entry of such terms. The approach described here was expanded and used successfully by other projects in our laboratory.[17,18]

Construction of the Dictionary

A key problem we encountered in this project was the development of the dictionary to be used in the matching process (translation of sentences into canonical forms) and in the programmatic grammar-generation process (substitution of synonyms). We experimented with automatic collection of synonyms for concepts from an on-line general (nonmedical) dictionary. The on-line dictionary was set up as a

database, which included for each entry corresponding definitions, senses, and a list of synonyms, antonyms, and related words. We extracted programmatically for each word its morphologic variants, senses, synonyms, and related words from the general on-line dictionary. We informally evaluated the usefulness of the resulting dictionary by using it to support the programmatic construction of grammars. The resulting grammars were extremely large, some even failed to compile by the SSI system. For the grammars that did compile, the recognition rate they produced was very poor. With these results, and after realizing that most of the words that we obtained from the general dictionary were not medically relevant, we abandoned our attempts to extract words programmatically from the general dictionary and used synonyms from a medical dictionary that was provided to us by the creators of QMR.

Comparison of Grammars

The evaluation experiment[3] demonstrated that when the SSI system used manually generated grammars, speech-recognition accuracy was greater and the matcher recognized more terms in QMR correctly than when the system used programmatically generated grammars. The most likely explanation for this difference relates to the accuracy of specification and to the expressiveness provided by the different grammars.

The manually constructed grammars were tailored more accurately than the programmatically constructed grammars to what one of us (WMD) perceived as the expected target language. Although the programmatically constructed grammars were independent of idiosyncratic expressions for findings, they allowed production of many sentences physicians would never say. For example, a physician would never say *the artery is carotid under a continuous bruit* to describe a bruit in the carotid artery, but this sentence was generated by the programmatically constructed grammar for findings related to the neck. The programmatically constructed grammars placed a heavy load on the speech-recognition system because the number of competing sentences that the system had to check was much larger than was required for interacting with the diagnostic system.

Misrecognition could partly be explained by the limited scope of the language covered by both grammars. First, for both grammars it was easy to find sentences that were likely inputs but that were not represented by the grammar. Second, both grammars did not represent nonlexical sounds that are commonly part of speech, such as pauses, coughs, and repetitions. Any effort to enlarge the scope of grammars must balance the need to include more valid sentences with the need to limit the number of possible matches to support good recognition accuracy. Expansion of grammars to include nonlexical words is currently not possible with the SSI system because the vocabulary recognized by the system does not include nonlexical sounds.

The performance gain from using manually generated grammars needs to be weighed against the time and training required to construct these grammars. On average, it took a medically trained developer five hours to construct a grammar that represented the QMR physical examination findings for a particular body part such as the chest. This time included defining the target language for that body part, entering grammar rules into the speech system, and testing whether common utterances would be captured by the grammar. For this experiment, we created grammars for seven body parts, so the total time required to produce the grammars approached 40 hours. Building grammars for all possible body parts would have taken two or three times as long.

In contrast, it took a software engineer approximately 80 hours to develop and implement the automated methods for grammar construction, but once the methods were developed, it took negligible time to generate a grammar. Although this approach took more preparation time for this experiment, it will be more time efficient in the long run because new, modified, or expanded grammars could be generated programmatically. Thus, as such systems are scaled up and are used for domains with changing vocabularies, the time to construct grammars would not increase dramatically.

Our conclusion from this experience with grammar construction is that manually generated grammars are superior when used for small domains that have stable vocabularies. However, for systems that require a large number of grammars or that need to represent languages that are constantly changing, an automated method may be preferable, especially if performance can be improved. One area for future research is devising ways to improve the precision of these automated grammars.

We believe that we can improve the identification accuracy by expanding the synonym dictionary to include more synonyms, and by extending the linguistic forms that are represented in the grammars. The addition of more synonyms would support correct identification of matches between input utterances and their corresponding controlled-vocabulary terms. Both modifications would promote accurate

representations of what users might say to the program and would increase the likelihood of accurate speech recognition.

Summary

Our program demonstrated the feasibility of spoken entry of terms into a medical application. However, the identification of QMR terms from spoken utterances was not sufficiently accurate to allow for use of our interface in a clinical setting. The large number of recognition errors we observed in the evaluation indicates that the grammars we constructed were too large despite the fact that we partitioned the domain of physical examination findings into the smaller subdomains of body parts. The errors also indicated that the grammars did not represent the spoken sentences accurately enough. We could produce more accurate grammars manually by collecting data systematically from a large number of potential users. We might be able to improve the programmatically generated grammars by having the program generate grammar rules and then testing the rules for semantic appropriateness to eliminate implausible sentences.

Improvements in speech-recognition technology are crucial for the development of speech interfaces that could be used reliably outside the laboratory. We expect that as speech-recognition technology improves so that less-restricted languages can be recognized, our pattern-matching approach to the interpretation of input utterances will still be effective to support interactions with knowledge-based decision support systems.

The authors thank Jeremy Wyatt, Charles Friedman, Kevin Johnson, Alex Poon, and Blackford Middleton for their support; Randolph Miller for providing findings from the INTERNIST-1 knowledge base; Camdat Corporation for providing synonym dictionaries derived from QMR; Lyn Dupre and Nora Sweeny for editing earlier versions of this article; and Edward Shortliffe, Edward Feigenbaum, and Bill Meisel for helping to establish the Speech Project at the Knowledge Systems Laboratory of Stanford University.

References ■

1. Feldman CA, Stevens D. Pilot study on the feasibility of a computerized speech recognition charting system. Community Dent Oral Epidemiol. 1990;18:213–5.
2. Bergeron B, Locke S. Speech recognition as a user interface. MD Comput. 1990;7(5):329–34.
3. Detmer WM, Shiffman S, Wyatt JC, Friedman CP, Lane CD, Fagan LM. A continuous-speech interface to a decision support system: II. An evaluation using a Wizard-of-Oz experimental paradigm. J Am Med Informatics Assoc. 1995;2:46–57.
4. Shiffman S, Wu AW, Poon AD, et al. Building a speech interface to a medical diagnostic system. IEEE Expert. 1990;6:41–50.
5. Wulfman CE, Rua M, Lane CD, Shortliffe EH, Fagan LM. Graphical access to medical expert systems: V. Integration with continuous-speech recognition. Methods Inf Med. 1993;32:33–46.
6. Miller RA, Masarie FE. Quick Medical Reference (QMR): an evolving microcomputer-based diagnostic decision-support program for general internal medicine. In: Proceedings of the Thirteenth Symposium on Computer Applications in Medical Care, Washington, DC, 1989:947–8.
7. Miller RA, Pople HE, Myers JD. INTERNIST-1: an experimental computer-based diagnostic consultant for general internal medicine. N Engl J Med. 1982;307:468–76.
8. Sager N, Friedman C, Lyman MS. Medical Language Processing: Computer Management of Narrative Data. Reading, MA: Addison-Wesley, 1987.
9. Evans DA. Concept management in text via natural-language processing: the CLARIT approach. In: Working Notes of the 1990 AAAI Symposium on Text-Based Intelligent Systems, Stanford University, Stanford, California, 1990:93–5.
10. Hersh WR, Greenes RA. SAPHIRE: an information retrieval system featuring concept matching, automatic indexing, probabilistic retrieval, and hierarchical relationships. Comput Biomed Res. 1990;23:410–25.
11. Schröder M. Supporting speech processing by expectations: a conceptual model of radiological reports to guide the selection of word hypotheses. In: Proceedings of the German Conference on Processing Natural Language. Nürnberg: University of Erlangen, 1992:7–9.
12. Lee KF. Automatic Speech Recognition: The Development of the SPHINX System. Boston: Kluwer Academic Publishers, 1989.
13. Jackson E, Appelt D, Bear J, Moore R, Podlozny A. A template matcher for robust natural language interpretation. In: Proceedings of Speech and Natural Language, Pacific Grove, California. San Mateo, CA: Morgan Kaufmann, 1991:190–3.
14. Young S, Hauptmann AG, Ward WH, Smith ET, Werner P. High level knowledge sources in usable speech recognition systems. Communications of the ACM. 1989;32(2):183–94.
15. Barr A, Feigenbaum EA. Understanding natural language. In: Barr A, Feigenbaum EA, eds. The Handbook of Artificial Intelligence. Los Altos, CA: William Kaufmann, 1981.
16. Salton G. Automatic Text Processing. Reading, MA: Addison-Wesley, 1990.
17. Johnson K, Poon AD, Lin R, et al. A history-taking system that uses continuous speech recognition. In: Proceedings of the Sixteenth Symposium on Computer Applications in Medical Care, Baltimore, Maryland, 1992;757–61.
18. Lin R, Lenert L, Middleton B, et al. A free-text processing system to capture physical findings: Canonical Phrase Identification System (CAPIS). In: Proceedings of the Fifteenth Symposium on Computer Applications in Medical Care, Washington, DC, 1991:168–72.