US 20020032564A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2002/0032564 A1

Ehsani et al. (43) Pub. Date: Mar. 14, 2002

(54) **PHRASE-BASED DIALOGUE MODELING WITH PARTICULAR APPLICATION TO CREATING A RECOGNITION GRAMMAR FOR A VOICE-CONTROLLED USER INTERFACE**

(76) Inventors: **Farzad Ehsani**, Los Altos Hills, CA (US); **Eva M. Knodt**, La Honda, CA (US); **Demitrios L. Master**, Cupertino, CA (US)

Correspondence Address:
David R. Graham
1337 Chewpon Avenue
Milpitas, CA 95035 (US)

(21) Appl. No.: **09/840,005**

(22) Filed: **Apr. 19, 2001**

Related U.S. Application Data

(63) Continuation-in-part of application No. 09/580,059, filed on May 27, 2000. Non-provisional of provisional application No. 60/198,402, filed on Apr. 19, 2000.

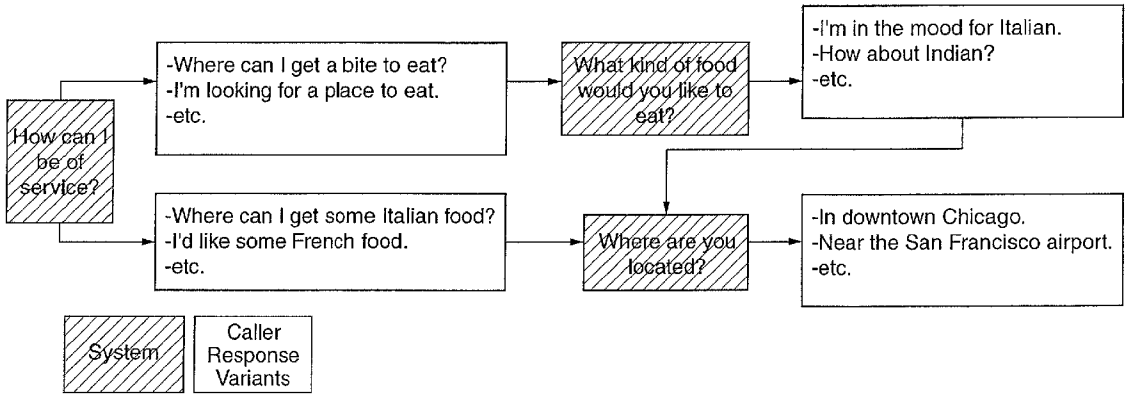
Publication Classification

(51) **Int. Cl.**[7] **G10L 15/12**; G10L 15/26; G10L 15/00; G10L 21/00

(52) **U.S. Cl.** **704/235**; 704/270.1; 704/275

(57) **ABSTRACT**

The invention enables creation of grammar networks that can regulate, control, and define the content and scope of human-machine interaction in natural language voice user interfaces (NLVUI). The invention enables phrase-based modeling of generic structures of verbal interaction to be used for the purpose of automating part of the design of such grammar networks. Most particularly, the invention enables such grammar networks to be used in providing a voice-controlled user interface to human readable text data that is also machine-readable (such as a Web page, a word processing document, a PDF document, or a spreadsheet).



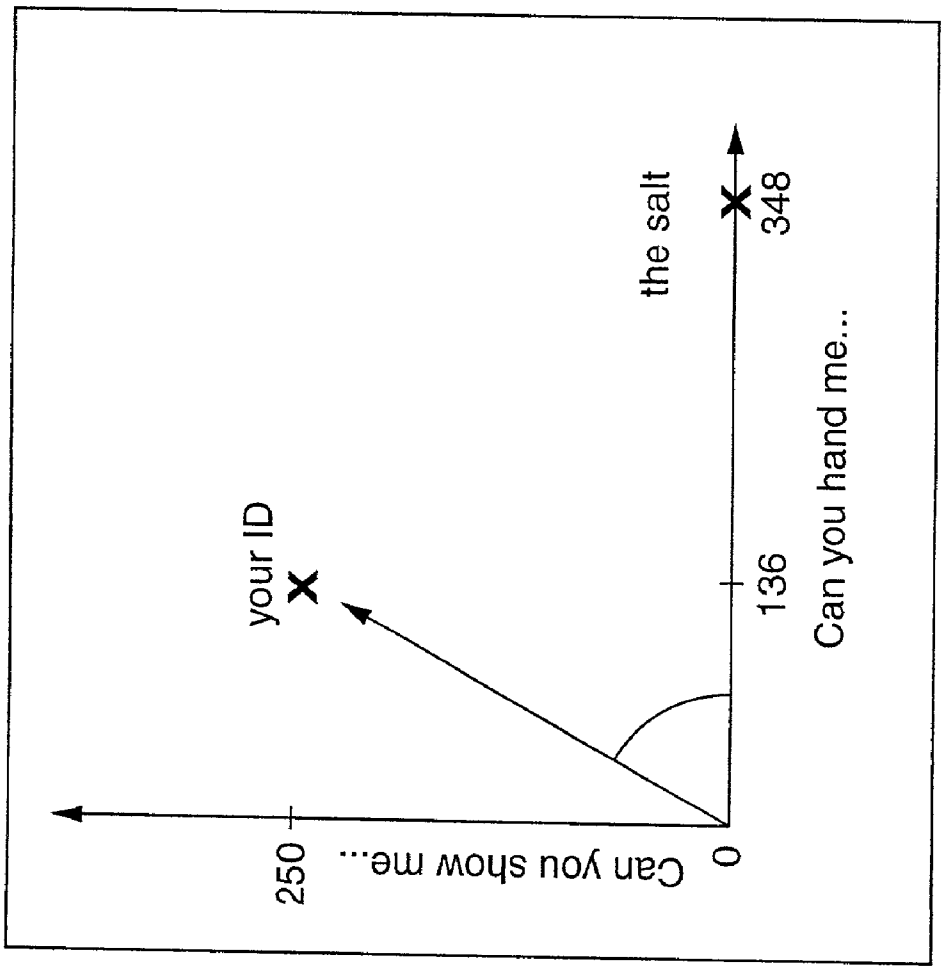


FIG. 1

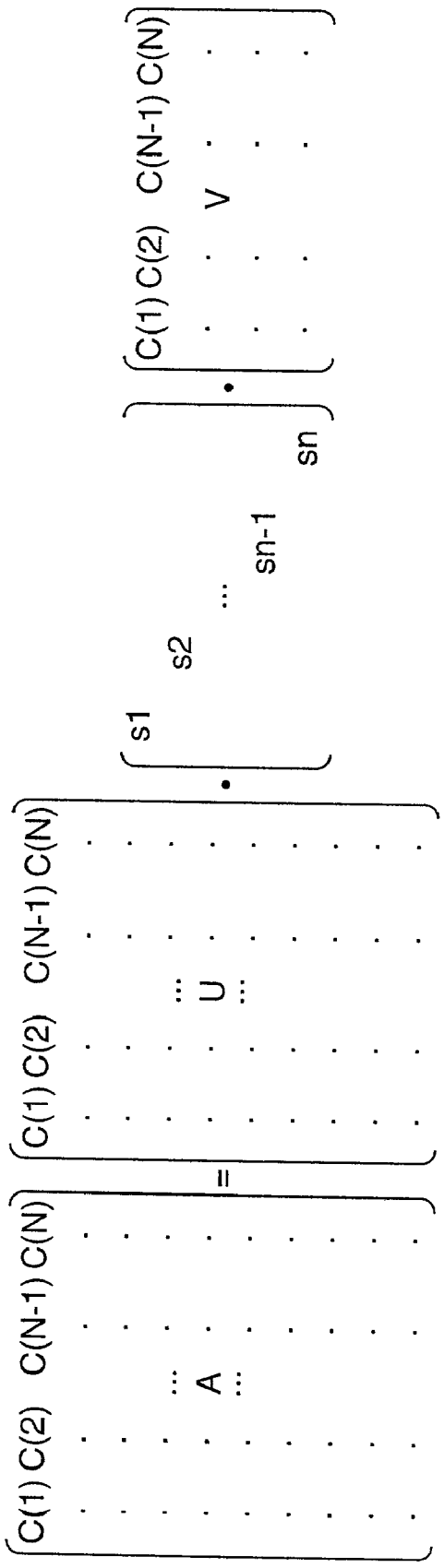


FIG. 2

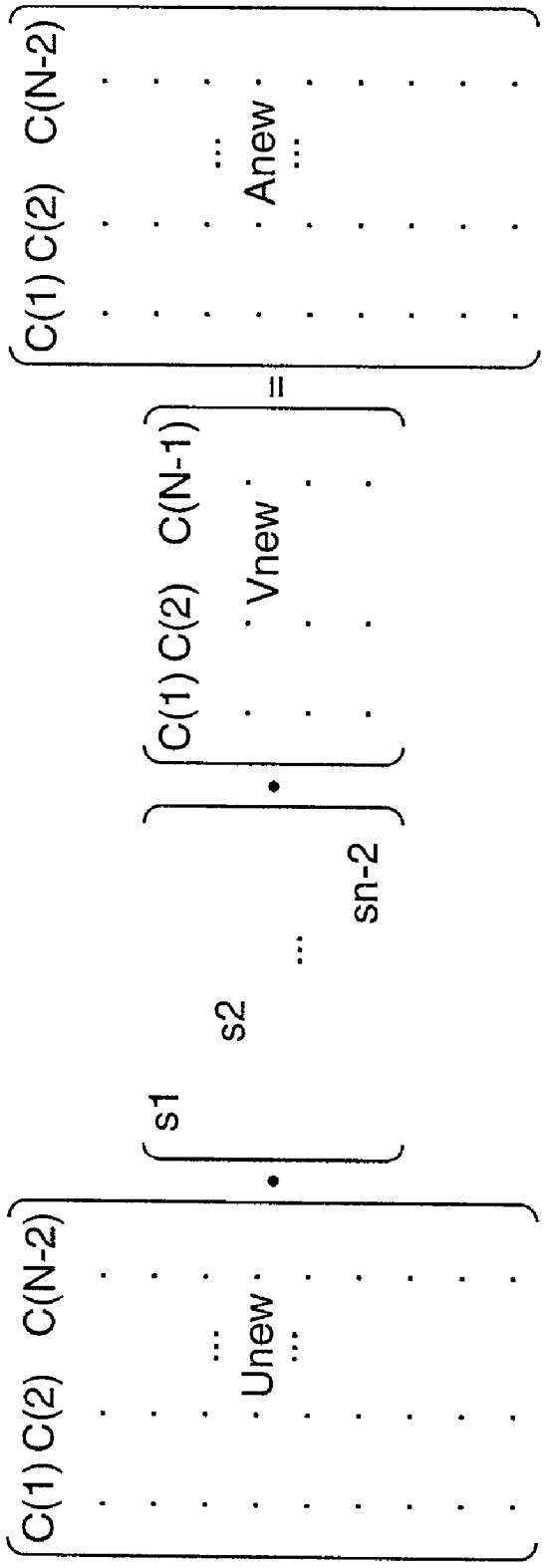


FIG. 3

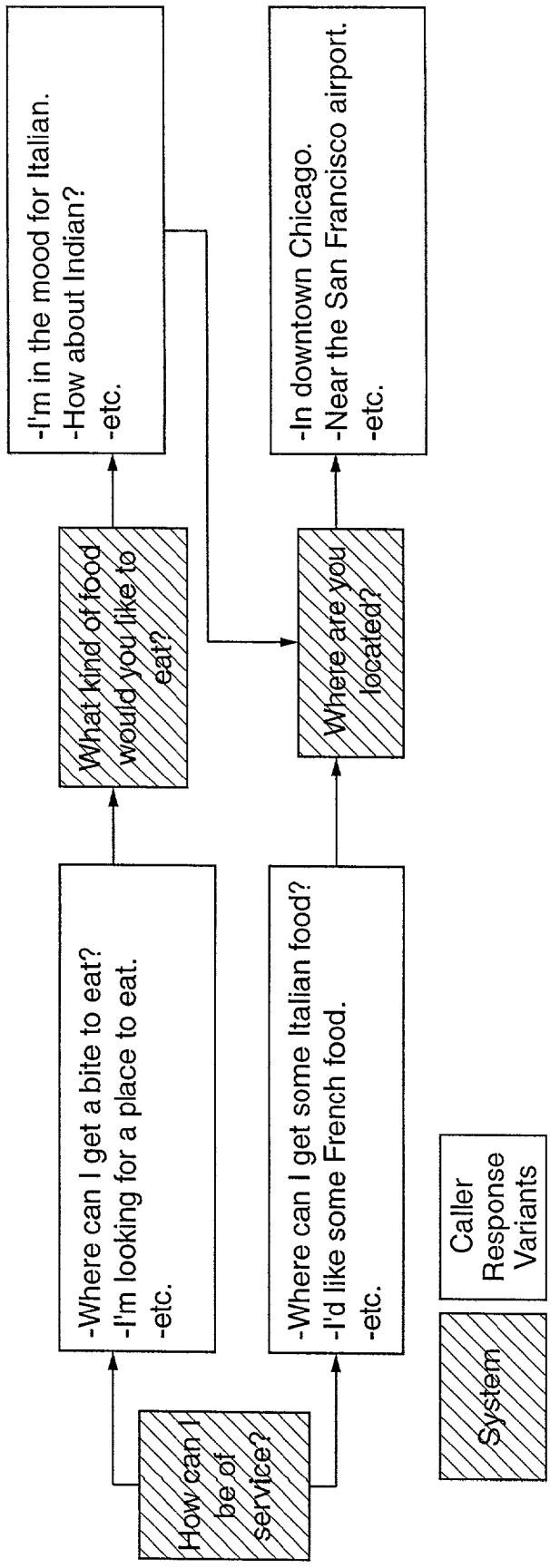

How can I be of service?
-Where can I get a bite to eat?
-I'm looking for a place to eat.
-etc.
-Where can I get some Italian food?
-I'd like some French food.
-etc.
What kind of food would you like to eat?
Where are you located?
-I'm in the mood for Italian.
-How about Indian?
-etc.
-In downtown Chicago.
-Near the San Francisco airport.
-etc.
System
Caller Response Variants

FIG. 4

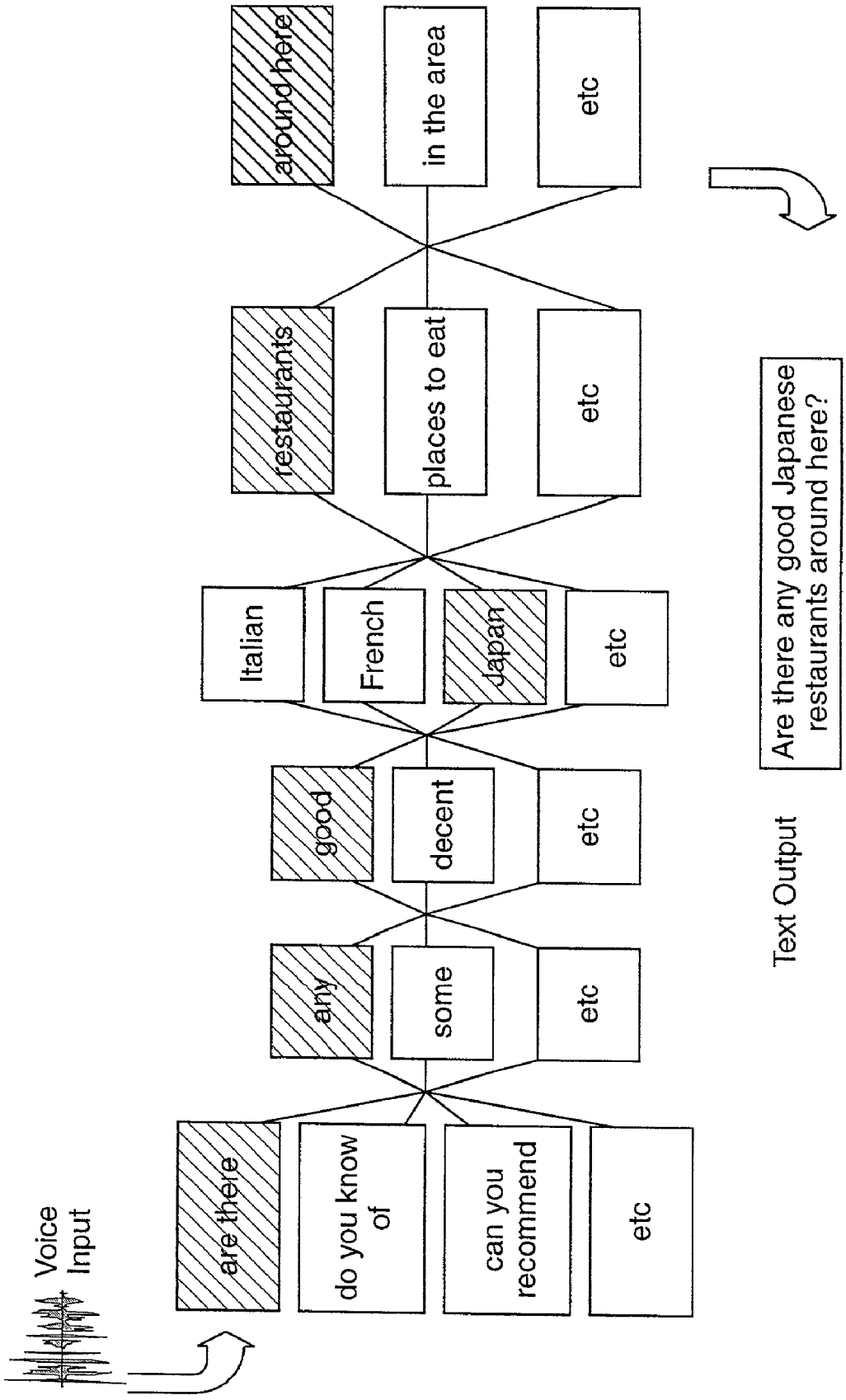

Voice Input
are there
do you know of
can you recommend
etc
any
some
etc
good
decent
etc
Italian
French
Japan
etc
restaurants
places to eat
etc
around here
in the area
etc
Text Output
Are there any good Japanese restaurants around here?

FIG. 5

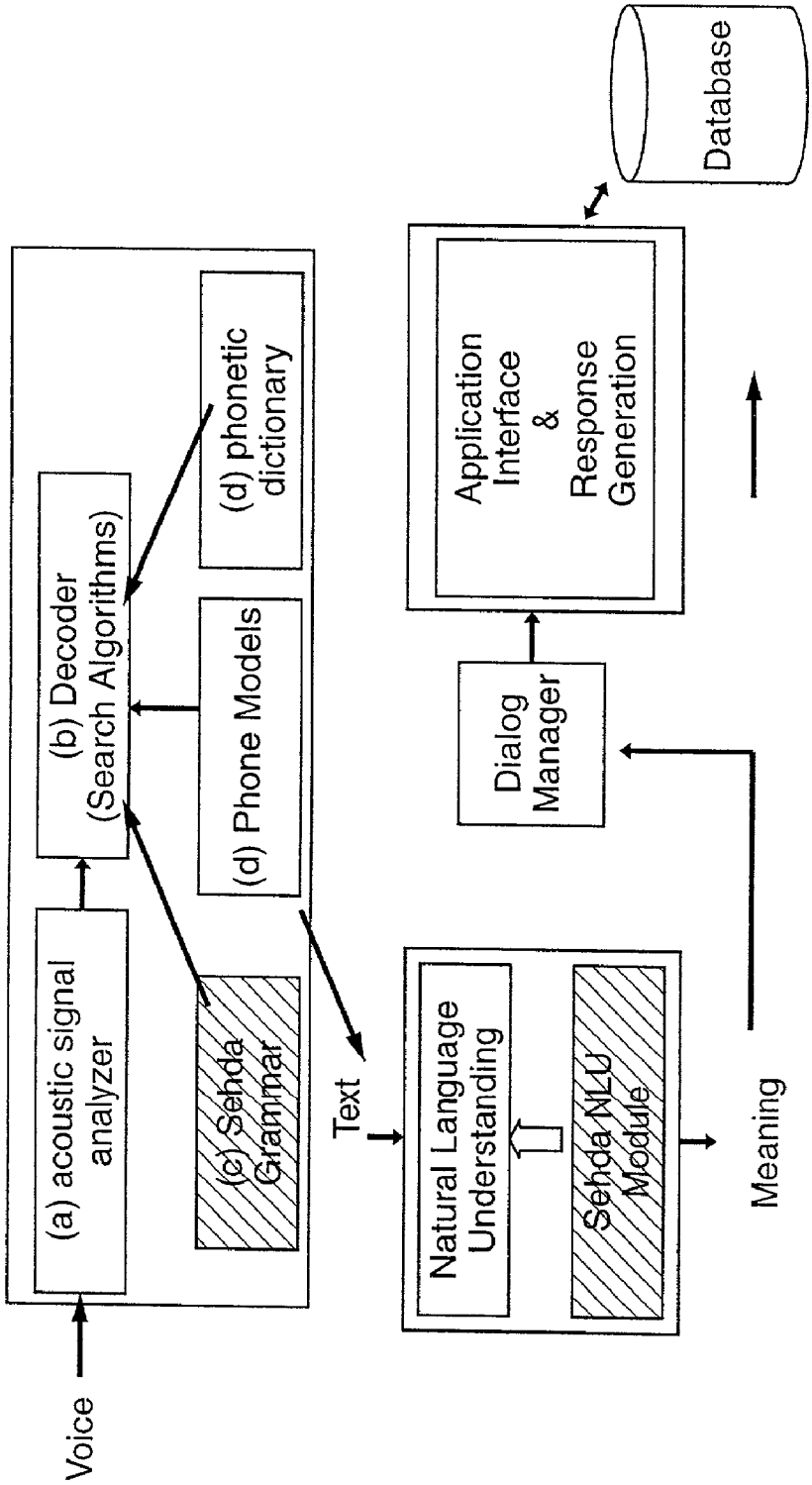

Database
Application Interface & Response Generation
(d) phonetic dictionary
(b) Decoder (Search Algorithms)
(d) Phone Models
(a) acoustic signal analyzer
(c) Sehda Grammar
Dialog Manager
Text
Natural Language Understanding
Sehda NLU Module
Meaning
Voice

FIG. 6

PHRASE-BASED DIALOGUE MODELING WITH PARTICULAR APPLICATION TO CREATING A RECOGNITION GRAMMAR FOR A VOICE-CONTROLLED USER INTERFACE

BACKGROUND OF THE INVENTION

[0001] 1. Field of the Invention

[0002] This invention relates to the creation of grammar networks that regulate, control, and define the content and scope of human-machine interaction in natural language voice user interfaces (NLVUI). More particularly, the invention relates to phrase-based modeling of generic structures of verbal interaction and use of these models for the purpose of automating part of the design of such grammar networks. Most particularly, the invention relates to the use of such grammar networks in providing a voice-controlled user interface to human readable text data that is also machine readable (such as a Web page, a word processing document, a PDF document, or a spreadsheet).

[0003] 2. Related Art

[0004] Voice user interfaces enable control of devices via voice commands transmitted through a microphone or telephone handset and decoded by a speech recognizer. These interfaces supplement or replace conventional input modalities such as a keyboard or a telephone touch-tone pad, and are increasingly deployed in a wide range of situations, where keyboard input is either inconvenient or impossible, e.g., to control home appliances, automotive devices, or applications accessed via the telephone. In recent years, a number of routine over-the-phone transactions such as voice dialing and collect call handling, as well as some commercial call center self-service applications, have been successfully automated with speech recognition technology. Such systems allow users to remotely access, for example, a banking application or ticket reservation system, and to retrieve information or complete simple transactions by using voice commands. Increasingly, voice control is being deployed to access the Internet by phone for the purpose of retrieving information or completing Internet-based commercial transactions such as making an on-line purchase.

[0005] a. Limitations and unsolved problems in current technology

[0006] Current technology limits the design of voice-controlled user interfaces in terms of both complexity and portability. Systems must be designed for a clearly defined task domain, and users are expected to respond to system prompts with short, fixed voice commands. Systems typically work well as long as vocabularies remain relatively small (200-500 words), choices at any point in the interaction remain limited and users interact with the system in a constrained, disciplined manner.

[0007] There are two major technological barriers that need to be overcome in order to create systems that allow for more spontaneous user interaction: (1) systems must be able to handle more complex tasks, and (2) the speech interface must become more "natural" if systems are expected to perform sophisticated functions based on unrestrained, natural speech or language input.

[0008] A major bottleneck is the complexity of the recognition grammar that enables the system to recognize natural language voice commands, interpret their meaning correctly, and respond appropriately. As indicated above, this grammar must anticipate, and thus explicitly spell out, the entire virtual space of possible user requests and/or responses to any given system prompt. To keep choices limited, the underlying recognition grammars typically process requests in a strictly predetermined, menu-driven order.

[0009] Another problem is portability. Current systems must be task specific, that is, they must be designed for a particular domain. An automated banking application cannot process requests about the weather, and, conversely, a system designed to provide weather information cannot complete banking transactions. Because recognition grammars are designed by hand and model domain specific rather than generic machine-human interaction, they cannot be easily modified or ported to another domain. Reusability is limited to certain routines that may be used in more than one system. Such routines consist of subgrammars for yes-no questions or personal user data collection required in many commercial transactions (e.g., for collecting name, addresses, credit card information, etc.). Usually, designing a system in a new domain means starting entirely from scratch.

[0010] Even though the need for generic dialogue models is widely recognized and a number of systems claim to be portable, no effective and commercially feasible technology for modeling generic aspects of conversational dialogue currently exists.

[0011] b. Current system design and implementation

[0012] The generated dialogue flow and the recognition grammar can be dauntingly complex for longer interactions. The reason is that users always manage to come up with new and unexpected ways to make even the simplest request, and all potential input variants must be anticipated in the recognition grammar. Designing such recognition grammars, usually by trained linguists, is extremely labor-intensive and costly. It typically starts with a designer's guess of what users might say and requires hours of refinement as field data is collected from real users interacting with a system simulation or a prototype.

[0013] c. Stochastic versus rule-based approaches to natural language processing

[0014] Since its beginnings, speech technology has oscillated between rule-governed approaches based on human expert knowledge and those based on statistical analysis of vast amounts of data. In the realm of acoustic modeling for speech recognition, probabilistic approaches have far outperformed models based on expert knowledge. In natural language processing (NLP), on the other hand, the rule-governed, theory-driven approach continued to dominate the field throughout the 1970's and 1980's.

[0015] In recent years, the increasing availability of large electronic text corpora has led to a revival of quantitative, computational approaches to NLP in certain domains.

[0016] One such domain is large vocabulary dictation. Because dictation covers a much larger domain than interactive voice-command systems (typically a 30,000 to 50,000 word vocabulary) and does not require an interpretation of the input, these systems deploy a language model rather than a recognition grammar to constrain the recognition hypotheses generated by the signal analyzer. A language model is

computationally derived from large text corpora in the target domain (e.g., news text). N-gram language models contain statistical information about recurrent word sequences (word pairs, combinations of 3, 4, or n words). They estimate the likelihood that a given word is followed by another word, thus reducing the level of uncertainty in automatic speech recognition. For example, the word sequence "A bear attacked him" will have a higher probability in Standard English usage than the sequence "A bare attacked him."

[0017] Another domain where probabilistic models are beginning to be used is automated part-of-speech analysis. Part-of-speech analysis is necessary in interactive systems that require interpretation, that is, a conceptual representation of a given natural language input. Traditional part-of-speech analysis draws on explicit syntactical rules to parse natural language input by determining the parts of an utterance and the syntactic relationships among these parts. For example, the syntactical rule S →NP VP states that a sentence S consists of a noun phrase NP and a verb phrase VP.

[0018] Rule-based parsing methods perform poorly when confronted with syntactically ambiguous input that allows for more than one possible syntactic representation. In such cases, linguistic preferences captured by probabilistic models have been found to resolve a significant portion of syntactic ambiguity.

[0019] Statistical methods have also been applied to modeling larger discourse units, such as fixed phrases and collocations (words that tend to occur next to each other, e.g. "eager to please"). Statistical phrase modeling involves techniques similar to the ones used in standard n-gram language modeling, namely, collecting frequency statistics about word sequences in large text corpora (n-grams). However, not every n-gram is a valid phrase: for example, the sequence "the court went into" is a valid 4-gram in language modeling, but only "the court went into recess" is a phrase. A number of different methods have been used to derive valid phrases from n-grams, including syntactical filtering, mutual information, and entropy. In some cases, statistical modeling of phrase sequences has been found to reduce lexical ambiguity. Others have used a phrase-based statistical modeling technique to generate knowledge bases that can help lexicographers to determine relevant linguistic usage.

[0020] Experiments in training probabilistic models of higher-level discourse units on conversational corpora have also been shown to significantly reduce the perplexity of a large-vocabulary continuous speech recognition task in the domain of spontaneous conversational speech. Others have modeled dialogue flow by using a hand-tagged corpus in which each utterance is labeled as an IFT (illocutionary force type). Probabilistic techniques have also been used to build predictive models of dialogue structures such as dialogue act sequences. The bottleneck in all of these experiments is the need for hand-tagging both training and testing corpora.

[0021] Another recent application of a probabilistic, phrase-based approach to NLP has been in the field of foreign language pedagogy, where it has been proposed as a new method of teaching foreign languages. Michael Lewis, in his book, *Implementing The Lexical Approach* (Hove, Great Britain, 1997) challenges the conventional view that learning a language involves two separate cognitive tasks: first, learning the vocabulary of the language, and second, mastering the grammatical rules for combining words into sentences. The lexical approach proposes instead that mastering a language involves knowing how to use and combine phrases in the right way (which may or may not be grammatical). Phrases, in Lewis' sense are fixed multi-word chunks of language, whose likelihood of co-occurring in natural text is more than random. Mastering a language is the ability of using these chunks in a manner that produces coherent discourse without necessarily being rule-based.

SUMMARY OF THE INVENTION

[0022] In one aspect, the present invention concerns modeling generic aspects of interactive discourse based on statistical modeling of phrases in large amounts of conversational text data. It involves automatically extracting valid phrases from a given text corpus, and clustering these phrases into syntactically and/or semantically meaningful equivalent classes. Various existing statistical and computational techniques are combined in a new way to accomplish this end. The result is a large thesaurus of fixed word combinations or phrases, grouped in equivalence classes that contain similar phrases. This thesaurus provides a data structure in which variations of saying the same thing and their associated probabilities can be looked up quickly. To the extent that this phrase thesaurus groups similar or semantically equivalent phrases into classes along with probabilities of their occurrence, it contains an implicit probabilistic model of generic structures found in interactive discourse, and thus can be used to model interactions across a large variety of different contexts, domains, and languages.

[0023] In another aspect of the invention, the phrase thesaurus mentioned above functions as a key element of a software application that can be used to generate recognition grammars for voice-interactive dialogue systems. The thesaurus provides the linguistic knowledge necessary to automatically expand anticipated user responses into alternative linguistic variants.

[0024] in another aspect of the invention, the phrase thesaurus is used as part of a software application that can be used to generate recognition grammars from the source code of a web page or pages, including "interactive" part(s) of web page(s) (i.e., part(s) of web page(s) that prompt the user to provide textual information in form fields) and/or "non-interactive" part(s) of web page(s) (i.e., part(s) of web page(s) other than interactive parts, such as parts of web page(s) that enable navigation within a web page and/or between web pages). For interactive part(s) of a web page, the software application takes the form-field keyword(s) provided in the page source, constructs an interactive dialogue flow based on the sequence of keyword(s), and automatically generates recognition grammars for the anticipated user responses. For informational (non-interactive) part(s) of a web page, the software application can use the phrase thesaurus to automatically generate recognition grammars for identified headings or topics within the web page. This aspect of the invention can be used generally to generate a recognition grammar from any set of human readable text data that is also machine readable. Though Web pages are an important example of such text data with which the invention can be used, the invention can also be

used with other such types of text data, such as text data created using a word processing program, PDF documents and text data created using a spreadsheet.

[0025] The present invention has a number of significant advantages over existing techniques for designing voice recognition grammars. Most significantly, it automates the most laborious aspects of recognition grammar design, namely, the need to generate, either by anticipation or by empirical sampling, potential variants of responses to any given system prompt. Secondly, it eliminates the need for expensive user data collection and hand coding of recognition grammars. Thirdly, the invention allows developers without specialized linguistic knowledge to design much more complex networks than conventional design techniques can support. In sum, the invention enables a developer to create more complex and better performing systems in less time and with fewer resources.

[0026] In another aspect of the invention, a compiled subset of the thesaurus (containing only the phrases incorporated into any given recognition grammar) is incorporated into a natural language understanding (NLU) component that parses the recognizer output at run-time to derive a conceptual meaning representation. Because phrases consist of words in context, they are potentially less ambiguous than isolated words. Because a phrase-based parser can draw on the linguistic knowledge stored in a large probabilistic phrase thesaurus, it is able to parse utterances much faster and with higher accuracy than conventional rule-based parsers.

BRIEF DESCRIPTION OF THE DRAWINGS

[0027] **FIG. 1** illustrates a two-dimensional vector space for the phrases "can you show me . . ." and "can you hand me." . . .

[0028] **FIG. 2** illustrates a matrix representation of a singular value decomposition algorithm.

[0029] **FIG. 3** illustrates a simplified matrix representation of a singular value decomposition algorithm.

[0030] **FIG. 4** is an example of a dialogue flow chart for a simple restaurant information request.

[0031] **FIG. 5** shows a type of recognition grammar for user responses to the system prompt: "What kind of food would you like to eat?"

[0032] **FIG. 6** illustrates the place of the present invention within an application that is controlled by a voice-interactive natural language user interface.

DETAILED DESCRIPTION OF THE INVENTION

[0033] I. Phrase-Based Dialogue Modeling

[0034] The present invention can enable a person with no special linguistic expertise to design a dialogue flow for an interactive voice application. It can be used to automatically generate a recognition grammar from information specified in a dialogue flow design. The key element in the present invention is a large, machine readable database containing phrases and other linguistic and statistical information about dialogue structures. This database provides the linguistic knowledge necessary to automatically expand a call-flow design into a recognition grammar. The following is a description of the components of the invention, how they are generated and how they work together within the overall system.

[0035] a. Phrase Thesaurus

[0036] The phrase thesaurus is a large database of fixed word combinations in which alternative ways of saying the same thing can be looked up. The phrases are arranged in the order of frequency of occurrence, and they are grouped in classes that contain similar or semantically equivalent phrases. The following is an example of a class containing interchangeable ways of confirming that a previous utterance by another speaker has been understood:

[0037] I understand

[0038] I hear you

[0039] [I] got [you ¦ your point ¦ it]

[0040] I see your point

[0041] I [hear ¦ see ¦ know ¦ understand][what you're saying ¦what you mean]

[0042] I follow you

[0043] [I'm ¦ I am] with you [there]

[0044] I [hear ¦ read] you loud and clear

[0045] Example Based On Michael Lewis, *Implementing The Lexical Approach: Putting Theory into Practice,* Howe, Great Britain, 1997.

[0046] The database comprises anywhere from 500,000 and 1 million phrase entries. The number of phrases may vary, depending on the size of the initial text corpus and the domain to be modeled. The minimum requirement is that the initial text corpus is large enough for statistical modeling. Generally, a larger, semantically richer corpus tends to yield a larger phrase database, which in turn is likely to provide a greater number of linguistic variants for each phrase.

[0047] In addition to the phrase entries, the database comprises a vocabulary of lexical items containing objects, locations, proper names, dates, times, etc. that are used to fill the slots in phrase templates such as "how do I get to . . .?" Some partial phrases may occur in several different groupings. For example, the sub-phrase "I know" in "I know what you mean" may also occur in another class containing alternate ways of challenging a speaker:

[0048] [I know ¦ I'm sure ¦ I believe] you're [wrong ¦ mistaken]

[0049] As a result, some phrase classes may be overlapping or contain cross-references between partial phrases.

[0050] b. Building a phrase thesaurus

[0051] The phrase thesaurus is generated automatically by a series of computer programs that operate on large amounts of natural language text data. The programs are executed sequentially, each taking the output of the previous program as its input, and processing it further. Taken together, the programs take a large text corpus as their input, and output a phrase thesaurus of the type described in section a. above. Some of the steps involved in this process are based on standard algorithms that have been used in various aspects of computational linguistics to process large machine read-

able corpora. These algorithms are used and combined within the present invention in a new way to accomplish the goal of automatically deriving a phrase thesaurus.

[0052] c. Linguistic assumptions underlying the invention

[0053] The present invention makes the following linguistic assumptions:

[0054] 1. Language in general, and conversational speech in particular, consists of phrases rather than of isolated vocabulary items, the combination of which is governed by grammatical rules.

[0055] 2. A phrase is a fixed, multi-word chunk of language of an average length between 1 and 7 words that conveys a unique sense depending on just that particular combination of words. The words that make up a phrase may or may not occur next to each other (e.g., the phrase "to make sense" can be separated by "a whole lot of,""not much," etc.)

[0056] 3. The use of phrases is governed by conventions of usage and linguistic preferences that are not always explicable with reference to grammatical rules. The phrase "on the one hand" loses its unique phrasal sense if "hand" is replaced by "finger.""On the one finger" is not a legitimate phrase in Standard English, even though it is perfectly grammatical. Being able to use just the right phrases signals native fluency in a speaker.

[0057] 4. There are at least four types of phrases:

[0058] (classification based on Lewis, 1997 and Smadja, 1994). The typology is not meant to be exhaustive or complete; other classifications may be possible.

[0059] (a) Polywords: generally 1-3 word fixed phrases conveying a unique idiomatic sense. Polywords allow for no variation or reversal of word order.

[0060] Example: "by the way,""nevertheless,""bread and butter,""every now and then."

[0061] b) Collocations: words that occur next to each other in more than random frequencies and in ways that are not generalizable: Example: "perfectly acceptable,""stock market slide,""sales representative." Variation in collocations is possible, but restricted by linguistic usage: "a tall building, ""a tall boy" (but not: "a high building,""a high boy"); "to take a look at a problem" (not: "to gaze at a problem"); "anxiety attack" (not "fear attack"), but also an "asthma attack," a "hay-fever attack."

[0062] (c) Standardized, idiomatic expressions with limited variability, often used in formulaic greetings and social interaction routines:

[0063] Example: "Howls it going?""How are you doing?""Thanks, I'm fine [great ¦ terrific]."Talk to you later."

[0064] (d) Non-contiguous phrases: functional frames containing one or more slots that can be filled by a limited number of words. The meaning of the phrase is determined by the filler word. The set of legitimate filler words tends to be determined by world knowledge rather than linguistic usage.

[0065] Example: "Can you pass me the . . . , please?" Here, the filler can be any small object that can be "passed on" by hand: "salt,""pepper,""bread,""water," but not "house, ""tree,""sewing-machine," etc. "I have a . . . in my shoe" can be filled by, e.g., "stone,""pebble,""something" , but not by "elephant."

[0066] 5. Because they are fixed in the mental lexicon of the speakers of the language, some word combinations are more likely to be observed/chosen in actual discourse than other combinations. This is why usage patterns and their frequencies can be analyzed using statistical methods, and can be captured in probabilistic models that reveal these patterns.

[0067] 6. Phrases are relatively unambiguous in their meaning or intention. Ambiguity arises when an utterance can have more than one conceptual meaning. The source of ambiguity can be either lexical (a word can have two or more unrelated meanings. E.g., "suit"=1. a piece of clothing, 2. a legal dispute), syntactic (a sentence can have two or more different and equally plausible parses (e.g. "he killed the man with a knife," where the modifier "with a knife" can either refer to VP (the act of killing) or to the NP (the object of killing). Because phrases use words in context, they reduce semantic ambiguity (wearing a suit vs. filing a suit) and some cases of syntactic ambiguity.

[0068] 7. Phrasal usage is not an exclusive property of spoken, conversational language. Rather, phrase usage pertains to all forms and genres of spoken and written discourse. However, each of these genres may use different types of phrases, and a computational analysis of linguistic preferences in terms of phrase frequencies and probabilities is likely to reveal different patterns of usage depending on the genre.

[0069] 8. Nor is phrasal usage an exclusive property of English. Most languages are governed by it, albeit in different ways. Generally speaking, phrases do not translate word for word into other languages. A literal translation, for example, of "get your act together" into German yields a meaningless construct "bring deine Tat zusammen." However, many phrases have functional phrase equivalents in other languages, e.g., "getting one's act together"=> "sich zusammenreiBen."

[0070] d. Goals of the invention

[0071] The following are goals of the present invention:

[0072] 1. To implement a phrase-based, corpus driven natural language processing technique that can reveal overarching discourse patterns without requiring laborious hand-tagging of training data in terms of syntactic, semantic, or pragmatic utterance features. As Lewis puts it: "Grammar tends to become lexis as the event becomes more probable" (p. 41). That is to say, syntactic, semantic, and

pragmatic structures are embedded in the phrase and are modeled along with it, provided the analysis is based on a conversational speech corpus large enough for statistical modeling.

[0073] 2. To implement the process described under 1) above in such a way that the resulting linguistic knowledge can be stored in a machine readable database, and used (and reused repeatedly) in a computer system designed to generate recognition grammars for voice-controlled user interfaces.

[0074] 3. To implement the process described under 1) above in such a way that the resulting linguistic knowledge can be stored in a machine readable database, and used (and reused repeatedly) in a Natural Language Understanding component that functions within a speech recognition system to extract the meaning of user responses at runtime.

[0075] e. Data Resources

[0076] Statistical modeling of any kind requires a vast amount of data. To build a sizable phrase thesaurus of 500,000 to 1 million entries requires a large source corpus (on the order of 1 billion words). However, smaller and more specialized corpora may be used to model phrases in a particular domain. For a phrase thesaurus covering the domain of interactive discourse, a number of diverse resources may be used to compile a text corpus for language. Such resources include but are not limited to:

[0077] 1. Transcribed speech databases for task oriented interactive discourse (SWITCHBOARD, CallHome, and TRAINS (available from the Linguistic Data Consortium (LDC) at www.ldc.upenn.edu).

[0078] 2. User data collected from verbal interactions with existing dialogue systems or with simulations of such systems.

[0079] 3. Closed caption data from television programs containing large amounts of interactive dialogue, such as talk shows, dramas, movies, etc. Television transcripts tend to be highly accurate (95%-100% for off-line captioned programs) (Jensema, 1996). As a consequence, virtually unlimited amounts of data can be purchased from places that gather and disseminate this data.

[0080] Television transcripts are a good way of supplementing databases of task-oriented discourse (1. and 2.) Even though most television shows are scripted, they nonetheless contain large amounts of common dialogic structures, good idiomatic English, etc. What is missing is mainly the fragmented, discontinuous nature of most conversational speech. However, this difference may well be an advantage in that models based on well-formed conversational speech might be used to identify and repair elliptical speech.

[0081] f. Data Preparation

[0082] To prepare the corpus for phrase modeling, it is subjected to a normalization procedure that marks sentence boundaries, identifies acronyms, and expands abbreviations, dates, times, and monetary amounts into full words. This normalization process is necessary because the phrase thesaurus is used to create grammars for recognition systems, and recognizers transcribe utterances as they are spoken, not as they are written. This means that monetary amounts, e.g., $2.50, must be spelled out in the recognition grammar as "two dollars and fifty cents" in order to be recognized correctly. The procedure also eliminates non-alphanumeric characters and other errors that are often found in television transcripts as a result of transmission errors in the caption delivery.

[0083] The normalization process is carried out by running a sequence of computer programs that act as filters. In the normalization process, raw text data is taken as input and a cleaned-up, expanded corpus that is segmented into sentence units is output. Sentence segmentation is especially important because the subsequent phrase modeling procedure takes the sentence as the basic unit.

[0084] The invention can make use of a version of a text normalization toolkit that has been made freely available to the speech research community (Copyright 1994, University of Pennsylvania, available through the Linguistic Data Consortium).

[0085] g. Compiling a seed dictionary of phrase candidates

[0086] The first step and the precondition for building a phrase thesaurus from a corpus is a creating a seed dictionary of likely phrase candidates. Initially, existing on-line idiomatic dictionaries are searched for basic phrase candidates that are rigid and not subject to grammatical or lexical variation (section I.c.4.(a)-(c)). The words and phrases are compiled into a basic phrase list. Less rigid collocations and phrasal templates are subject to considerable lexical and grammatical variability, and therefore, empirical text data are needed that contain actual instances of their use. To compile an initial seed phrase dictionary, we derive collocations automatically from large corpora on the basis of simple frequency counts, and then subject the results to a post-processing heuristics to eliminate invalid collocations.

[0087] Step 1: Deriving N-Grams

[0088] We begin by deriving n-gram statistics from a given corpus C1 using standard language modeling techniques. For an overview of such techniques, see Frederik Jelinek, Frederick, *Statistical Methods for Speech Recognition,* MIT, Cambridge Mass., 1997). The procedure generates information about how often word strings of n-word length occur in a given corpus. Input: A given Corpus C1→ Output: n-gram frequency counts.

[0089] We choose n-grams of varying lengths (approximately 1<=n<=7.) N-grams are sorted in the order of the frequency of their occurrence.

[0090] Step 2: Filtering: Deriving Valid Phrase Candidates From N-Grams

[0091] The list of n-grams is very large and contains many invalid and meaningless collocations, phrase fragments, and redundant word combinations that are subsumed by larger n-grams.

[0092] Take for example, the following sentence: "<s> e-mail is replacing to a large extent direct communication between people </s>."

[0093] For 1 <= n <=7, n-gram frequency counts on this sentence, including sentence boundary markers, will return 70 unique n-grams (13 unigrams, 12 bigrams, 11 trigrams,

10 4-grams, 9 5-grams, 8 6-grams, and 7 7-grams). By contrast, the sentence contains only four potentially valid phrase candidates, two of which are partially overlapping:

[0094] (a) Phrase template: "replacing [. . .] communication"

[0095] (b) Multi-word: "to a large extent"

[0096] (c) Compound noun collocation: "direct communication"

[0097] (d) Mixed collocation: "communications between people"

[0098] The next step consists of filtering n-grams to eliminate invalid or redundant collocations by implementing a series of computational measures to determine the strength of any given collocation. The problem of n-gram filtering can be approached in a number of different ways, and the following description is meant to be exemplifying rather than being exhaustive. Since the goal at this point is to compile a preliminary seed dictionary of phrases, any of the methods described below can be used, either by themselves or in combination, to identify initial phrase candidates.

[0099] A Frequency-Based Pre-Filtering Method

[0100] The simplest filtering method is frequency-based. Computed over a large corpus, n-grams with high frequency counts are more likely to contain strong collocations than n-grams that occur only once or twice. We eliminate n-grams below a specific frequency threshold. The threshold is lower for large word strings because recurring combinations of large n-grams are rarer, and more likely to contain significant phrase candidates than shorter strings.

[0101] Perplexity/Entropy

[0102] Perplexity is a measure for determining the average branching factor of a recognition network and it is most often used as a measure for evaluating language models. It indicates the probability, computed over an entire network, that any given element can be followed by any other. For example, in a digit recognition system composed of 0-9 digits and two pronunciations for 0 ("oh" and "zero"), the perplexity of the recognition grammar exactly equals the number of elements, 11, because there are no constraining factors that favor certain digit sequences over others. Because word sequences underlie various kinds of constraints (imposed by syntax, morphology, idiomatic usage, etc.) perplexity has been found useful in natural language processing to measure the strength of certain collocations (see, for example, Shimohata, S, T. Sugio, J. Nagata, "Retrieving Collocations by Co-occurrence and Word Order Constraints," Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, 1997, pp. 476-481.)

[0103] We take each unique n-gram and its associated frequency f(n-gram) and look at the probability of each word w_i that can follow the n-gram. We calculate this probability $p(w_i)$ by dividing the frequency in which a given word follows the n-gram by the frequency count for the n-gram itself:

[0104]

$$p(W_1) = \frac{f(wi)}{f(n\text{-gram})}$$

[0105] If the n-gram is part of a larger, strong collocation, the choice of words adjacent to the phrase boundary will be very small, because of the internal constraint of the collocation. Conversely, the likelihood that a particular word will follow is very high. For example, the word following the trigram "to a large" will almost always be "extent," which means, the perplexity is low, and the trigram is subsumed under the fixed collocation "to a large extent." On the other hand, a large number of different words can precede or follow the phrase "to a large extent," and the probability that any particular word will follow is very small (close to 0).

[0106] We use a standard entropy measure to calculate the internal co-locational constraints of the n-gram at a given junction wi as:

$H(\text{n-gram})=\Sigma -p(w_j)\ln p(w_1)$

[]=wordj]

[0107] The perplexity of the n-gram can then be defined as:

$\text{Prep(n-gram)}=e^{H(\text{n-gram})}$

[0108] We eliminate n-grams with low surrounding perplexity as redundant (subsumed in larger collocations) and keep the ones with perplexity above a specified threshold t.

[0109] Step 3: Deriving Non-Contiguous Phrases

[0110] The frequency and perplexity measures described above give us a good first cut at phrase candidates, generating mainly rigid word combinations such as compound nouns ("Grade Point Average"), idiomatic expressions ("How's it going?") and polywords ("sooner or later"). The next objective is to expand the initial seed phrase dictionary by deriving non-contiguous collocations (collocations that are less rigid and contain one or more filler words or phrases, e.g. "Give me . . . please"). There are at least three types of non-contiguous phrases. Assuming that w is any word and p is any phrase, these types can be distinguished as follows:

[0111] Type 1: $p_1 \ldots . P_2$

[0112] Two phrases occurring next to each other with more than random frequency, separated by one or more words that are not themselves phrases.

[0113] Example: "refer to [the appendix ¦ the manual ¦ page 220. . .] for more information"

[0114] Type 2: $p_1 \ldots w_1$ A phrase is followed or preceded by one or more filler words, which are followed or preceded by another word that, together with the initial phrase, forms a phrase template.

[0115] Example: "Could you hand me [the salt ¦ your ID . . .] please?"

[0116] Type 3: $w_1 \ldots . W_2$ A word is followed by one or more filler words, which are followed by another word that together with the initial word forms a phrase template.

[0117] Example: "taking [initial ¦ the first ¦ important . . .] steps"

[0118] To extract phrases of the types 1 and 2, we first create a list of contexts for each phrase. We take each of the phrase candidates obtained in the first processing phase and retrieve all sentences containing the phrase. We then look at surrounding words in order to identify possible regularities and co-occurrence patterns with words or phrases not captured in the initial n-gram modeling and filtering stage. This can be done using any of the following methods: frequency counts, normalized frequency methods, perplexity, or normalized perplexity.

[0119] In order to handle Type 3, we compile a list of the top n most frequent word bigrams separated by up to 5 words. As in the first extraction stage, not every collocation is significant. Again, there are several ways to eliminate invalid collocations that can be used by themselves or in various combinations. Again, this can be done using any of the following methods: frequency counts, normalized frequency methods, perplexity, or normalized perplexity.

[0120] Mutual Information

[0121] Mutual information is a standard information theoretical measure that computes the strength of a relationship between two points by comparing the joint probability of observing the two points together with the probability of observing them independently. In natural language processing, it has been used to establish the strength of an association between words, for example, for use in lexicography (see Kenneth Church, W. & Patrick Hanks, "Word Association Norms, Mutual Information, and Lexicography,"*Computational Linguistics,* 16 (1), 1990: 22-29.)

[0122] Given two phrases, ql and q2 with probabilities p(q1) and p(q2) then the mutual information I (ql, q2) is defined as:

$$I(q1, q2) = \frac{p(q_1, q_2)}{p(q_1)p(q_2)}$$

[0123] Joint probability can serve as a measure to determine the strength of a collocation within a given window (in our case, a sentence), even if the collocation is interrupted, as in the case of non-contiguous phrases. If there is a genuine association between two words or word strings, their joint probability will be larger than the probability of observing them independently, so the mutual information I(w1,w2) must be greater than 1.

[0124] We take our corpus of non-contiguous phrase candidates and compute the mutual information for each phrase and the most frequent words or word sequences surrounding these phrases. We extract the phrase-word or phrase-phrase combinations with the highest joint probability.

[0125] However, the above formula may generate misleading results in case of very frequently used words such as "the,""it," or "very good." In this case we will use a slightly modified mutual information defined as:

$$I_{new}(q1, q2) = \frac{p(q_1, q_2)}{p(q_1)}$$

[0126] where q2 is the frequent word or phrase.

[0127] Probability Distribution

[0128] Yet another way to eliminate invalid phrase candidates is to look at the probability distribution of components within each non-contiguous phrase candidate. For each phrase candidate, we determine a main component and a sub-component (the longer or the more frequent phrases can usually be considered as the main component), and then look at the probability distribution of the sub-component with respect to other words or phrases that co-occur in the same context (i.e., sentence or clause). This algorithm can be formally described as:

$$M_{main,sub} = \frac{f(q_{main}, q_{sub}) - \mathrm{Exp}(q_{main})}{Dev(q_{main})}$$

[0129] where $f(q_{main}, q_{sub})$is the frequency of the co-occurrence of the main component with the sub-component and $Exp(q_{main})$ & $Dev(q_{main})$ are the Expected Value and the Standard Deviation of the frequency occurrence of q_{main} with all of the sub-components q_{sub}.

[0130] We can assume that if $M_{main,\ sub}$ is greater than a certain threshold, then the collocation is a valid phrase, otherwise it is not.

[0131] Hand Checking

[0132] A final way of eliminating invalid phrases—especially cases determined as borderline by the other algorithms—is by having a trained linguist go through the resulting phrase dictionary and eliminate the unlikely phrases. This step, while optional, may improve the quality and accuracy of the resulting phrase list with respect to common linguistic usage.

[0133] Step 4: Phrase-Based Corpus Segmentation

[0134] As explained in the previous section, a number of measures can be (and have been) used to automatically derive an initial seed dictionary of phrase candidates from large corpora. Because all of these methods act more or less as filters, they can be used in various combinations to extract multi-word phrases and collocations. However, whatever method we use, the list of derived phrases still contains a large number of overlapping phrase candidates, because multiple parses of the same sentence remain a possibility. For example, for the sentence "E-mail is replacing direct communications between people," the following alternative parses are conceivable:

[0135] Parse 1: <s>[E-mail][is replacing][direct communications][between people] </s>

[0136] Parse 2: <s>[E-mail][is replacing direct communications][between people] </s>

[0137] Parse 3: <s>[E-mail][is replacing][direct] [communications between people.] </s>

[0138] The problem is similar to the one we encounter when segmenting text for building dictionaries in Chinese or Japanese. In these languages, the concept of a "word" is less well defined than it is in European languages. Each Chinese word is made up of anywhere between one and seven characters, and in Chinese writing, word boundaries are not separated by white spaces. The problem is augmented by the fact that complete Chinese dictionaries are extremely hard to find, especially when it comes to proper names.

[0139] The absence of word boundaries in Chinese or Japanese creates significant difficulties when building probabilistic language models for large vocabulary dictation systems. Word-based n-gram language modeling requires correct parsing of sentences to identify word boundaries and subsequently calculate n-gram probabilities. Parsing errors are a common problem in Chinese language processing. For example, we may encounter a character sequence ABCDE where A, AB, CDE, BCD, D, and E are all legitimate words in the dictionary. One can quickly note that there are two possible parses for this character sequence: [A][BCD][E] and [AB][CDE]. Linguists have applied various lexical, statistical, and heuristic approaches, by themselves and in combination, to parse Chinese text. Most of these methods can be applied to phrase parsing in English. We describe one statistical, n-gram-based parsing algorithm that we found particularly efficient and useful. However, other methods can be used for phrase parsing as well.

[0140] The general idea is to implement an N-gram phrase-based language model (a language model that uses phrases rather than single words as the basis for n-gram modeling), in order to calculate the best parse of a sentence. Note that some words may act as phrases as can be seen in Sentence 3 (e.g. the word "direct" in the above example). Assuming the log probability bigram statistics for the example above to be as follows:

[0141] [<s>], [Email]--5.8

[0142] [Email],[is replacing]--2.4

[0143] [Email],[is replacing direct communications]--6.5

[0144] [is replacing], [direct]--4.7

[0145] [is replacing], [direct communications]--5.4

[0146] [direct],[communication between people]--4.2

[0147] [direct communications],[between people]--6.2

[0148] [is replacing direct communications],[between people]--8.9

[0149] [between people][<s>]-4.8

[0150] [communication between people][<s>]--5.9 Given these log probabilities, we can calculate the best phrase-based parse through a sentence by multiplying the probabilities (or summing the log probabilities) of each of the bigrams for each possible parse:

[0151] Parse $1_{\text{Total likelihood}}$=−5.8+−2.4+−5.4+−6.2+−4.8=−24.6

[0152] Parse $2_{\text{Total likelihood}}$=−5.8+−6.5+−8.9+−4.8=−26.0

[0153] Parse $3_{\text{Total likelihood}}$=−5.8+−2.4+−4.7+−4.2+−5.9=−23.0

[0154] We select the parse with the highest overall likelihood as the best parse (in this case, Parse 1).

[0155] A First Pass At Phrase-Based N-Gram Parsing

[0156] In order to create a phrase-based parse of a given text corpus C, we need a phrase-based language model. Building such a language model, however, requires a pre-parsed text or a dictionary of phrases. In order to get around this problem, we use a bootstrapping technique that provides us with an initial parse of the corpus, which will then form the basis for building an initial language model that is subsequently refined by iterating the procedure. There are two ways to derive a preliminary parse through the corpus:

[0157] 1. We use a Greedy Algorithm that, whenever it encounters a parsing ambiguity (more than one parse is possible), selects the longest phrases (e.g., the parse that produces the longest phrase or the parse that produces the longest first phrase) from the seed dictionary. In the above example, Parse 2 would be selected as the optimal parse.

[0158] 2. We pick the parse that minimizes the number of phrases for each parse. Assuming that neither the phrase "is replacing direct communications" (because it is not a very common phrase) nor the word "direct" are in the seed dictionary, Parse 1 would be selected.

[0159] Applying either one or both of these algorithms will result in an initial phrase-based parse of our corpus.

[0160] Optimizing the phrase-based n-gram parse

[0161] Once we have an initial parse through our corpus, we divide the corpus into two sub-corpora of equal size, C1 and C2 and use the seed dictionary of phrases (described in section I.b.-d.) to build an initial language model for one of the sub-corpora. We then use this language model to generate an improved segmentation of the other sub-corpus C2. Resulting high-frequency bigrams and trigrams are phrase candidates that can be added to the dictionary for improved segmentation.

[0162] A significant advantage of using a language modeling technique to iteratively refine corpus segmentation is that this technique allows us to identify new phrases and collocations and thereby enlarge our initial phrase dictionary. A language model based corpus segmentation assigns probabilities not only to phrases contained in the dictionary, but to unseen phrases as well (phrases not included in the dictionary). Recurring unseen phrases encountered in the parses with the highest unigram probability score are likely to be significant fixed phrases rather than just random word sequences. By keeping track of unseen phrases and selecting recurring phrases with the highest unigram probabilities, we identify new collocations that can be added to the dictionary.

[0163] There are two ways of implementing this procedure. In the first case, we start a unigram language model,

and use this model to segment sub-corpus C2. The segmented sub-corpus C2 is subsequently used to build a new, improved unigram language model on the initial sub-corpus C1. We iterate the procedure until we see little change in the unigram probability scores. At this point we switch to a bigram language model (based on phrase pairs) and reiterate the language modeling process until we see very little change. Then we use a tri-gram model (based on sequences of three phrases) and reiterate the procedure again until we see little changes in the segmentation statistics and few new, unseen phrases. At this point, our dictionary contains a large number of plausible phrase candidates and we have obtained a fairly good parse through each utterance.

[0164] In the second case, we implement the same iterative language modeling procedure, using bigram, trigram, or even n-gram models with larger units, in the very beginning of the process rather than increasing gradually from unigram to trigram models. One or the other implementation may prove more effective, depending on the type of source material and other variables.

[0165] h. Automatically deriving a phrase thesaurus from a seed dictionary of phrases

[0166] The core of the proposed technology is a phrase thesaurus, a lexicon of fixed phrases and collocations. The thesaurus differs from the seed dictionary of phrases in that it groups phrases that are close in content and in some sense interchangeable. The grouping is essential for the use of the phrase database in the context of the proposed invention, namely, to allow for the retrieval of alternative phrase variants that can be used to automatically create a grammar network. We use linear algebra techniques to determine the semantic distance between phrases contained in our phrase dictionary. Once we have a measure of closeness/distance between phrases, we can use this information and a standard clustering algorithm (e.g., Group Average Agglomerative Clustering) to derive sets of semantically similar phrases.

[0167] Step 1: Measuring Distance Between Phrases

[0168] In order to derive a measure for determining semantic distance between phrases, we draw on two basic linguistic assumptions:

[0169] 1. The meaning of a word is determined by its use. Mastering a language is the ability to use the right words in the right situation.

[0170] 2. The degree of similarity between two words can be inferred from the similarity of the contexts in which they appear. Two words are synonymous if they are completely interchangeable in all contexts. Two words are similar if they share a subset of their mutual contexts.

[0171] We take these assumptions to hold true not only for isolated words, but for phrases as well. To determine semantic proximity or distance between phrases, we look at the surrounding words and phrases that co-occur with any given phrase P across an entire machine readable corpus C, and measure the extent to which these contexts overlap. For example, we will find that the phrases "can you hand me . . . " and "can you pass me . . . " share a large subset of neighboring words: "salt,""coffee,""hammer,""the paper, ""my glasses," etc. Conversely, we find no overlap in the neighbors of the phrases "can you pass me . . . " and "can you tell me . . ."

[0172] To represent and measure semantic and/or syntactic relationships between phrases, we model each phrase by its context, and then use similarities between contexts to measure the similarity between phrases. One can imagine that each phrase is modeled by a vector in a multi-dimensional space where each dimension is used for one context. The degree of overlap between vectors indicates the degree of similarity between phrases. A simple example illustrates how to represent contextual relationships between phrases and their associated neighbors in such a space. For the two phrases, P1: "can you hand me . . . " and P2: "can you show me . . . , " we create an entry in a 2 dimensional matrix for each time they co-occur with one of two right neighbors, "the salt," and "your ID." The example shows that the phrases P1 and P2 share some but not all of the same contexts. P1 occurs 136 times with "your ID" but never (0 times) with "the salt." P2 co-occurs 348 times with "the salt" and 250 times with your ID.

[0173] We can capture this co-occurrence pattern geometrically in a two-dimensional space in which the phrases P1 and P2 represent the two dimensions, and the contexts "the salt" and "your ID" represent points in this space (see **FIG. 1**). The context the salt is located at point 0,348 in this space because it never occurs (0 times) with P1 and occurs 348 times with P2.

[0174] The degree of similarity between contexts can be determined by using some kind of association measure between the word vectors. Association coefficients are commonly used in the area of information retrieval, and include, among others, the following: Dice coefficient, Jaccard's coefficient, Overlap coefficient and Cosine coefficient (for an Overview, see C. J. van Rijsbergen, *Information Retrieval,* 2nd ed., London, Butterworths, 1979). There is little difference between these measures in terms of efficiency, and several of these coefficients may be used to determine the difference between phrases. The most straightforward one is the Cosine coefficient, which defines the angle a between the two word vectors as follows:

$$\cos\Theta = \frac{A^T B}{\|A\| \cdot \|B\|}$$

[0175] Step 2: Singular Value Decomposition

[0176] Using either of the formulas described in Step 1 will give us an initial distance measure between phrases. Assuming the phrase dictionary derived so far contains N phrases (with N being anywhere from 500,000 to 1,000,000), and assuming further that we parameterize each key-phrase with only the most frequent M phrases (with M being between 500,000 and 100,000 depending on a number of variables), then we still have two problems:

[0177] 1. The resulting MxN matrix may be too large (500,000×100,000) to compare vectors.

[0178] 2. Because of the sparseness of data, many context phrases or words will not appear in the context of their respective key phrases. For less frequent phrases or context phrases, the vector model might therefore yield misleading and inaccurate results.

[0179] In order to get around both of these problems we can use Singular Value Decomposition (SVD) to reduce the original matrix to a smaller and informationally richer matrix. We describe the original matrix as follows: each row is used for one key-phrase and each column is used for one of the M context-phrases. So c_{ij} is the number of occurrences of the phrase p_j in the context of phrase p_i. The standard SVD algorithm for a matrix A of size MxN allows us to express A as a product of a MxN column-orthogonal matrix U, a diagonal matrix S of size NxN whose elements are either positive or zero, and transpose of another NxN row-orthonormal matrix V. This can be summarized as follows:

$$A=U\cdot S\cdot V^{T}$$

[0180] The shapes of these matrices can be visualized as a series of columns, as shown in **FIG. 2**.

[0181] The advantage of using SVD is that it allows us to break down the matrix into its individual components and to reduce the size of the matrix by as much as one order of magnitude by eliminating unwanted or meaningless components. If the matrix is singular, some of the s_n will be zero and some are going to be very small. By eliminating these elements and reducing the matrix in size, we can make the matrix smaller and more manageable. Moreover, the reduced matrix A_{new} contains only the most significant elements of the original matrix A. Assuming that the sn-was very small and s_n was zero and we decide to eliminate these columns from the original matrix, the result would be a (M)x(N-2) matrix made from the first N-2 columns of U, S, & V, as shown in **FIG. 3**.

[0182] Note that Factor Analysis or any other kind of Principle Component Analysis with dimensionality reduction might work just as well in this case.

[0183] Step 3: Phrase Clustering

[0184] The next step in creating a phrase thesaurus consists of clustering phrases into classes based on the degree of overlap between distance vectors. A number of standard clustering algorithms have been described in the literature. The most efficient ones include Single Link, Complete Link, Group Average, and Ward's algorithm. These algorithms are typically used to classify documents for information retrieval, and, depending on the particular data being modeled, one or the other has been shown to be more efficient. For a discussion of clustering algorithms, see, e.g., El Hamdouchi, A. and P. Willett, "Hierarchic Document Clustering using Ward's Method," Proceedings of the Organization of the 1986 ACM Conference on Research and Development in Information Retrieval, 1988, pp. 149-156; El Hamdouchi, A. and P. Willett, "Comparison of Hierarchic Agglomerative Clustering Methods for Document Retrieval," The Computer Journal 32.3, 1989, pp. 220-227; Cutting, Douglas, R., David R. Krager, Jan O. Pedersen, John W. Tukey, "Scatter/Gather: A C1uster-Based Approach to Browsing Large Document Collections," Proceedings of the 15th Annual International SIGIR '92, Denmark, pp. 318-329.

[0185] All of these clustering algorithms are "agglomerative" in that they iteratively group similar items, and "global" in that they consider all items in every step.

[0186] We can use one or the other of these algorithms to cluster similar phrases into equivalence classes by performing the following steps:

[0187] a) Calculate all inter-phrase similarity coefficients.

[0188] Assuming q_x and q_y are any two phrases, they can be represented by rows X & Y of A_{new} from Step 2, so the similarity between any two phrases using the Cosine coefficient would be:

$$S_{\cos}(q_x, q_y) = \frac{q_x^T \cdot q_y}{\|q_x\| \cdot \|q_y\|}$$

[0189] b) Assign each phrase to its own cluster

[0190] c) Form a new cluster by combining the most similar pair of current clusters (r, s)

[0191] d) Update the inter-phrase similarity coefficients for all distances using r & s.

[0192] e) Go to step (c) if the total number of clusters is greater than some specified number N.

[0193] Clustering algorithms differ in how they agglomerate clusters. Single Link joins clusters whose members share maximum similarity. In the case of Complete Link, clusters that are least similar are joined last, or rather an item is assigned to a cluster if it is more similar to the most dissimilar member of that cluster than to the most dissimilar member of any other cluster. Group Average clusters items according to their average similarity. Ward's method joins two clusters when this joining results in the least increase in the sum of distances from each item to the centroid of that cluster.

[0194] Clustering techniques tend to be resource intensive, and some initial seeding of clusters, based on rough guesses, may be necessary. The Buckshot algorithm (Cutting, et. al., 1992) can be used to accomplish this goal. Buckshot starts with a small random number of clusters and then uses the resulting cluster centers (and just these centers) to find the right clusters for the other items. One could imagine other similar algorithms that take some initial guesses at the cluster center, and then use the cluster center (or even the top N items that can be considered as the closest to the center), and find the other buckets accordingly.

[0195] We can use any one of these clustering algorithms or a combination of them depending on the computational resources required and other factors to derive both flat and hierarchical groupings of phrases.

[0196] Step 4: Hand tagging of classes

[0197] In a final step, a sub-set of the hand-checked phrase classes are tagged with abstract descriptors denoting abstract conceptual representations of the phrases contained in each class. Descriptors include speech act classifications for verb phrases (e.g. request [. . .], confirm [. . .], reject [. . .], clarify [. . .], etc. and object nouns (e.g. date, location, time, amount,) and proper names (businesses, restaurants, cities, etc.).

[0198] The phrases in a phrase thesaurus produced in accordance with the invention can be arranged in a hierarchical manner. For example, phrases that can occur as part of other phrases can be represented once in the phrase thesaurus and each other phrase that can include such phrase

can include a pointer to that phrase. This can be desirable to enable the phrase thesaurus to be represented more compactly, thus decreasing the data storage capacity required to store the data representing the phrase thesaurus.

[0199] II. Use Of The Invention In Designing And Operating voice-interactive speech applications

[0200] Speech recognition technology is increasingly being used to facilitate communication between humans and machines in situations where the use of other input modalities (such as a keyboard) is either impossible or inconvenient. More specifically, such situations include remote access of databases and/or control of applications or devices using a telephone or other hand-held device and simple natural voice commands. Typically, callers dial into a voice telephony server and are led through a sequence of voice-driven interactions that lets them complete automated transactions such as getting information, accessing a database or making a purchase. Systems differ with regard to the complexity of the supported interaction and the manner in which the voice interface is integrated with the application it controls. In some cases, both the voice-interface and the application or back-end database are located on the same telephony server. In other cases, such as when telephone voice input is used to control Internet-based applications, the voice telephony server, which processes the telephone voice input, is linked with the application over the Internet.

[0201] In what follows, we describe three embodiments of the present invention and how they can be used to optimize both the design and the performance of speech applications:

[0202] 1. An application for designing recognition grammars for "generic" speech applications. By "generic" we mean that the grammars generated by means of this application can be used in a variety of different systems, such as computer desktop applications, remote voice control of household appliances, or telephone self-service applications.

[0203] 2. An application for designing recognition grammars for speech applications that allow Internet access by an audio input device (such as a telephone) and are therefore tightly integrated with the Internet. This application is similar to the immediately preceding embodiment, but, in addition, comprises features specifically designed for Internet-based audio input applications.

[0204] 3. A natural language component that functions as part of a speech application and extracts meaning from user responses at runtime.

[0205] In order to clarify the aspects of novelty inherent in these embodiments, the description of the embodiments is prefaced with a general overview of a standard speech application that illustrates how the grammar and the natural language understanding component function within the context of such applications. The present invention is particularly concerned with components **1** (e) and **2** in the description provided below.

[0206] The operation of a voice-interactive application entails processing acoustic, syntactic, semantic, and pragmatic information derived from a user's voice input in such a way as to generate a desired response from the application. This process is controlled by the interaction of at least five separate but interrelated components (see **FIG. 6**):

[0207] 1. a speech recognition front-end consisting of: (a) an acoustic signal analyzer, (b) a decoder, (c) phone models, (d) a phonetic dictionary, and (e) a recognition grammar;

[0208] 2. a Natural Language Understanding (NLU) component;

[0209] 3. a Dialogue Finite State Machine;

[0210] 4. an application Interface; and

[0211] 5. a speech output back-end.

[0212] The components enumerated above work together in the following manner:

[0213] 1. When a speech signal is received through a microphone or telephone hand-set, its acoustic features are analyzed by the acoustic signal decoder (a) and a set n of the most probable word hypotheses are computed based on the acoustic information contained in the signal, and the phonetic transcriptions contained in the dictionary (d). The dictionary is a word list that maps the vocabulary specified in the recognition grammar (e) to their phonetic transcriptions. The recognition grammar (e) defines legitimate user responses including their linguistic variants and thus tells the system what commands to expect at each point in a given interaction. Because the grammar specifies only legitimate word sequences, it narrows down the hypotheses generated by the acoustic signal analyzer to a limited number of possible commands that are can be recognized by the system at any given point. The result of the front-end processing is a transcription of the speech input.

[0214] 2. The Natural Language Understanding component (component 2) extracts the meaning of the transcribed speech input and translates the utterances specified in the recognition grammar into a formalized set of instructions that can be processed by the application. In most simple systems, this is done via language interpretation tags that are inserted manually into the grammar in such a way as to reduce the linguistic variants specified in a given recognition grammar to a single command that can be executed by the system. For example, the input variants "I'd like to order <title>,"Do you have <title>?," and "I'm looking for <title>" are reduced to a single instruction such as <search TITLE>.

[0215] 3. The Dialogue Finite State Machine (component 3) can be implemented as a computer program that specifies the flow of the human-machine interaction. It contains instructions for prompting the caller for speech input and for generating the appropriate system response to each instruction that is passed to the program by the natural language understanding component. The Dialogue Finite State Machine for a voice interface to an online bookseller's web site, for example, might prompt the user to say his/her name, address, credit card number, and upon successful completion of these items ask the user to say the title of the book he/she is looking for.

[0216] 4. The Application Interface can be implemented as a set of scripts that are called by the Finite State Machine and interact with the application that is controlled by the voice interface. These scripts contain instructions to be executed by the application (e.g., to access a bookseller's database and retrieve a requested title, to produce a verbal system response such as a request for clarification such as "Do you want Edgar Smith or Frank Smith?," or a combination of both). If the voice interface is used to control a web-based application, the specified instruction, e.g., <search TITLE>, is sent over the Internet from a voice server (discussed further below) to the respective web site where it is processed like a regular on-line transaction.

[0217] 5. The speech-output back-end (component 5) takes the verbal response generated by the application interface and maps it to an acoustic speech signal, using either a speech synthesizer or prerecorded utterances from a database.

[0218] (For a comprehensive overview of state-of-the-art dialogue systems, their operation, and assessment, see Ronald Cole, A. J. Mariani, Hans Uszkoreit, Annie Zaenen, Victor Zue, "Survey of the State of the Art in Human Language Technology, Center for Spoken Language Understanding," Oregon Graduate Institute, 1995, and EAGLES, *Handbook of Standards and Resources for Spoken Dialoque Systems,* De Gruyter, Berlin & New York, 1997.)

[0219] A. A Computer System For Automatically Creating Recognition Grammars For Voice-Controlled User Interfaces

[0220] The phrase thesaurus described above can be implemented as part of a computer system that can be used to automatically generate complex recognition grammar for speech recognition systems. The recognition grammar can then be used with an interactive user interface that is responsive to spoken input (voice input). The recognition grammar enables interpretation of the spoken input to the user interface. the system combines call-flow design, network expansion, and grammar compilation into a single development tool. The thesaurus forms the key element of this system, but in order to function in the manner desired, it must be integrated and work together with a number of other system components.

[0221] The system consists of the following components: (a) a graphical user interface for designing and editing the call flow for a voice application, (b) a network expander that retrieves alternative variants for the user commands specified in the call-flow design from the database along with their probabilities, (c) a linguistic database, (d) an editor, and (e) a compiler that translates the grammar network into a format than can be used by commercial speech recognizers.

[0222] (a) Call Flow Design: The first step in designing a recognition network for a voice-controlled dialogue system consists of specifying the call flow in such a way as to anticipate the logic of the interaction. The system's graphical user interface allows the designer to specify user requests, system states, and the transitions between these states. **FIG. 4** shows the initial part of a call flow for a simple restaurant information request. At this stage of the design process, the designer only needs to specify one sample utterance for each type of user request. For example, the utterance "Where can I find a good Japanese restaurant around here" fully specifies the request type "request restaurant information."

[0223] (b) Network Expander: In a second step, the user responses in the call flow design are automatically expanded into recognition grammars. A grammar includes the set of user responses to system prompts that the system can recognize and process accordingly. **FIG. 5** shows the type of network that needs to be generated to recognize the user response to the systems prompt "What kind of food do you like to eat?" For each user request, the grammar specifies the set of legitimate variants and supplies an abstract meaning representation (e.g., "request restaurant information"). Note that the system will not recognize speech input that is not explicitly specified in the grammar. If the recognition system allows for probabilistic grammars, the Network Expander can supply frequency and other probabilistic bigram and trigram statistics to build such a grammar.

[0224] Activation of the network expander will take the sample user responses specified in the call-flow design and automatically retrieve alternative linguistic variants from the database. For example, suppose we want to model a user request for help. For the phrase "I need help," the network expander will return: "What do I do now?,""Help!,""Help me, please,""I could need some help here!,""Can you help me?,""I'm lost, I don't know what to do,""Oops, something's wrong!," etc.

[0225] (c) Linguistic Database: The linguistic knowledge required for the automatic grammar expansion is stored in a large, machine-searchable database. The database contains the phrase thesaurus (along with probability scores associated with each phrase). In addition, it contains lists of common nouns for filling phrase templates, as well as locations, dates, proper names, etc. The database is customizable, that is, users can create their own application specific lists of objects, names, etc.

[0226] (d) Editor: The grammar designer provides editing functionality at all stages in the design process. Initial call flow designs can be saved, retrieved, and changed in both graphical and text mode. After the network has been expanded, the designer can go back to the initial call flow design and edit the phrase variants retrieved by the system. At this stage, most of the editing activity will consist of eliminating variants that don't fit the pragmatic context, and of completing phrase templates by accessing the supplemental databases provided by the system or by typing in the template fillers directly. The editor also permits review and modification of the meaning representations automatically supplied by the system.

[0227] (e) Compiler: After completing the editing, the user activates the system compiler, which executes a computer program that translates the grammar network design into a format that can be used by the recognizer.

[0228] In conventional grammar design, grammar network expansion must be done by hand. The knowledge of anticipated user responses and their linguistic variants is supplied by language experts who anticipate a set of variants to generate a grammar, or they are collected by recording user interactions with system simulations or prototypes. In accordance with the invention, grammar network expansion

can be automated using linguistic knowledge derived from previous modeling of linguistic behavior.

[0229] B. A Computer System For Creating Voice Interfaces For Internet-Based Self-Service Applications

[0230] Speech recognition technology can be used to enable access to the Internet by telephone or other audio input device to, for example, retrieve information or complete Web-based self-service transactions such as ordering tickets. A speech application located on a voice server (discussed further below) that is connected to the Internet allows callers to complete the same kinds of transactions they usually do via their Web-browser (e.g., register for a service, input credit card information, put together a shopping basket, or make a purchase). The difference from using a Web browser is that they use their voice rather than filling out interactive forms on the Web using a keyboard, mouse or other tactile input device. A voice server recognizes the voice input and sends data (e.g., a completed site registration or credit card transaction) to a Web site where it can be processed in the same way as a regular on-line transaction.

[0231] A "voice page" is a representation, e.g., a set of instructions and/or data (for convenience, sometimes referred to herein as "code"), of a conventional Web page that reproduces some or all of both the structure and content of the Web page, and enables interaction with the Web page using audio input—speech or tone(s) of predetermined pitch (e.g., DTMF). A voice page can include all of the five components of a speech application described above and shown in **FIG. 6**. Creating a voice page involves translating the graphical user interface (GUI) of the Web page, which is typically written in a markup language such as Hyper Text Markup Language (HTML) or extensible Markup Language (XML), into code for recognizing and processing voice commands. A voice page can be implemented in Voice XML (VXML), an HTML-like language for scripting a dialog flow and telephony interactions.

[0232] A "voice server" stores and enables access to "voice pages." For example, a voice server could be a software application running on a computer system which can be accessed by end-users via communication lines (e.g., telephone lines, Ti lines, ISDN lines, cable lines). In accordance with the invention, a voice server can be used as an interface for a web-based application. When used in such manner, the voice server can be adapted to transmit and receive data from the Internet. Alternatively, the voice server can be implemented together in the same apparatus as that used to implement the Web-based application. Further, when a voice server is used as an interface for a Web-based application, the voice page is closely integrated with the corresponding Web page as described above. If the voice page is implemented in VXML, an existing commercial voice server that works directly with VXML pages (e.g., Nuance Voice Web Server™) can be used.

[0233] Below, an aspect of the present invention is described in which phrase-based language processing is deployed to facilitate the translation of transaction-oriented Web pages into voice pages. Specifically, this aspect of the invention enables the generation of recognition grammars directly from information provided in the source code used to generate the corresponding Web page. This aspect of the invention is similar in functionality to the grammar design tool described above in that its key component is a phrase-based language-processing engine that supports automatic grammar expansion. In addition, however, the system comprises the following components:

[0234] 1. An off-the-shelf, or hand-built, HTML/XML parser (using conventional parsing technology) that extracts input field keywords from the Web page's source code; and

[0235] 2. A knowledge database that turns keywords into context (e.g., the "Title" field in an on-line bookseller's web site to "Please tell me the title of the book"). This knowledge database is used to create a Dialogue Finite State Machine.

[0236] Like the grammar design tool, the system provides a graphical interface for call-flow design and a large database of phrases for enabling the grammars to handle natural variations of user input, e.g., different ways of phrasing a request for information.

[0237] In one embodiment, the process of translating an interactive web page into a voice page comprises the following steps:

[0238] 1. A reference web page is loaded into memory and displayed on a computer screen.

[0239] 2. The page is parsed using the HTML/XML parser into HTML/XML tags (such as form tags).

[0240] 3. The tags can be grouped together into usable higher level modules (such as combining all of the form tags).

[0241] 4. An HTML parser can be implemented to search the source code of the web page for system prompts preceding form tags fields and itemized lists of option values for input into these fields. Additionally or alternatively (depending on the nature of the web page and/or the application for which the invention is used), the HTML parser can be implemented to search the source code of the web page for non-interactive parts of the web page (e.g., informational text and headings, navigation links that are internal or external to the web page). On the displayed web page, these items are marked in such a way that they can be clicked, copied, and pasted into the call-flow designer window.

[0242] 5. A call flow for the voice page can be designed by copying the items marked by the HTML parser into the call-flow designer window. These items include system prompts and lists of option values to be selected.

[0243] 6. System prompts are expanded into complete speech utterances as needed. This can be done manually by a designer or automatically by apparatus that accesses a linguistic knowledge base that contains expanded versions for system prompts commonly used in transaction-based web forms. For example, the "title" prompt in a booksellers on-line purchasing form can be expanded into the question: "What is the title of the book you are looking for?" The linguistic knowledge base can also be used to expand informational text (headings) and navigation links. For example, a list of book titles preceded by

a heading "Other books by the Author" can be expanded to "Would you like a list of other books the Author has written?"

[0244] 7. Lists of option values are expanded into phrases. Lists of option values can be automatically expanded into phrases using the same knowledge base described in step 6. For example, the option value for title can be translated into: "I'd like to buy <TITLE>," with <TITLE> being any of the titles that the user is allowed to buy at that point. Informational text (headings) and navigation links can be manually translated into questions by a designer selecting an appropriate phrase from a set of possible phrases presented to the designer. For example, the heading "Other books by the Author" can be changed to "What other books have been written by the author?"

[0245] 8. The phrase created in step 7 is expanded into a recognition grammar containing possible linguistic variants for the phrase "I'd like to buy <TITLE>."

[0246] 9. Call flow and grammars are edited using an editor of the type described above.

[0247] The grammars are compiled into a format that can be used by the voice recognizer.

[0248] Above, an embodiment of the invention is described in which a phrase thesaurus is used as part of a software application that can be used to generate recognition grammars from the source code of a web page or pages. The web page(s) can include "interactive" part(s) (i.e., part(s) that prompt the user to provide textual information in form fields) and/or "non-interactive" part(s) (i.e., part(s) other than interactive parts, such as part(s) that enable navigation). More generally, other embodiments of the invention similar to such an embodiment of the invention can be used to generate a recognition grammar from any set of human readable text data that is also machine readable, such as, for example, text data created using a word processing program, PDF documents, or text data created using a spreadsheet. Further, though the invention is often described above as implemented to generate a recognition grammar for text data representing a form, such need not be the case. The invention can also be used to generate recognition grammars based on text data that enables navigation through, or retrieval of information from, the set of text data. For example, the invention can be used to create a recognition grammar based on an index of this document. A voice recognition system could then be implemented to make use of such a recognition grammar so that, for example, when a user said "Skip to the claims," the voice recognition system would understand and act on that statement by the user.

[0249] C. A Natural Language Understanding Component To Be Used In Speech Recognition Systems

[0250] In another aspect of the invention a compiled sub-set of the phrase thesaurus is incorporated into a speech recognition system to be accessed at run-time in order to parse the incoming speech signal and to derive an abstract conceptual representation of its meaning that is passed on to the application. The phrase subset used in the run-time natural language interpreter is identical to the one used in a particular grammar. (Recall that the grammar specifies the total set of user commands the system expects and is able to process. Commands not specified in the grammar are automatically assigned to a single variable that triggers a system request for clarification.)

[0251] This aspect of the invention particularly concerns the NLU component. In conventional spoken dialogue systems, recognition grammars are mapped onto a set of formalized instructions by using a crude technique called "word spotting." Word spotting proceeds from a given set of instructions and then searches the user input for specific words that match these instructions. The instructions themselves are provided by hand, using abstract denominators to tag each grammar. Word spotting works by disregarding utterances or parts of utterances that are deemed irrelevant at a given state of the user-machine interaction. For example, all the responses specified in the request for a certain book title are reduced to the simple instruction "search <TITLE>," where only the words that make up "<TITLE>" are explicitly recognized by the system. Word spotting works for very simple systems, but it is limited by the fact that it cannot recognize negations or more complex syntactic relationships.

[0252] In the present invention, recognition grammars are mapped to system instructions by way of an annotation scheme that extracts the abstract meaning from a number of alternative phrase variants. This is possible because the underlying thesaurus database classifies phrases according to semantic similarity and contains pre-tagged descriptors for each class. At run-time, user speech input is parsed automatically into phrase-based units, which are subsequently translated into system instructions.

[0253] Various embodiments of the invention have been described. The descriptions are intended to be illustrative, not limitative. Thus, it will be apparent to one skilled in the art that certain modifications may be made to the invention as described herein without departing from the scope of the claims set out below.

We claim:

1. A method for creating a recognition grammar for use with an interactive user interface to human readable text data that is also machine readable, the interactive user interface being responsive to spoken input, the method comprising the steps of:

formulating an expression representing a part of the text data for each of one or more parts of the text data, wherein each formulated expression can be constructed as one or more combinations of one or more phrases in a phrase thesaurus; and

automatically using the phrase thesaurus to construct one or more equivalent expressions of one or more formulated expressions, wherein the recognition grammar comprises the collection of all of the expressions.

2. A method as in claim 1, wherein the step of formulating an expression representing a part of the text data further comprises the step of formulating an expression representing an interactive part of the text data.

3. A method as in claim 1, wherein the step of formulating an expression representing a part of the text data further comprises the step of formulating an expression representing a non-interactive part of the text data.

4. A method as in claim 1, wherein the text data represents one or more Web pages.

5. A method as in claim 4, wherein the step of formulating an expression further comprises the step of automatically parsing code representing the one or more Web pages to identify the one or more parts.

6. A method as in claim 5, wherein one or more of the parts of the one or more Web pages comprise a system prompt indicating a type of interaction with an interactive part of a web page and a plurality of option values each representing a possible input to the interactive user interface for that type of interaction, the step of parsing further comprising the step of identifying the system prompt and the plurality of option values.

7. A method as in claim 6, wherein the step of formulating an expression further comprises the steps of:

automatically identifying one or more phrases that correspond to a system prompt; and

automatically identifying, for each of a plurality of option values, one or more phrases that correspond to the option value.

8. A method as in claim 4, wherein the code representing the one or more Web pages is expressed in a markup language.

9. A method as in claim 8, wherein the code representing the one or more Web pages is expressed in HTML.

10. A method as in claim 8, wherein the code epresenting the one or more Web pages is expressed in XML.

11. A method as in claim 4, wherein the recognition grammar is expressed in VXML.

12. A method as in claim 1, wherein the step of automatically using the phrase thesaurus to construct one or more equivalent expressions further comprises the steps of:

selecting a combination of one or more phrases representing the formulated expression, wherein the phrases of the selected combination of one or more phrases are original phrases of the formulated expression;

identifying an equivalent phrase for each of one or more original phrases of the formulated expression; and

producing a new combination of one or more phrases representing the formulated expression, the new combination including at least one of the identified equivalent phrases, wherein the new combination represents the equivalent expression.

13. A method as in claim 12, wherein:

phrases in the phrase thesaurus have a probability of occurrence associated therewith;

one or more original phrases has a plurality of equivalent phrases; and

the step of identifying an equivalent phrase further comprises the step of selecting an equivalent phrase having the highest probability of occurrence.

14. A method as in claim 12, wherein equivalent phrases are grouped in classes and each class of equivalent phrases has associated therewith a descriptor denoting a conceptual representation of the phrases contained in that phrase class, the method further comprising the step of tagging each equivalent expression with the descriptor or descriptors associated with phrases of the equivalent expression.

15. A method as in claim 1, further comprising the step of translating the recognition grammar into a form that can be processed by a speech recognition system.

16. A method as in claim 1, further comprising the step of manually editing the recognition grammar.

17. A method as in claim 1, wherein expressions representing a plurality of parts of the text data are formulated and the phrase thesaurus is used to identify equivalent expressions for a plurality of formulated expressions.

18. A system for creating a recognition grammar for use with an interactive user interface to human readable text data that is also machine readable, the interactive user interface being responsive to spoken input, the system comprising:

means for formulating an expression representing a part of the text data for each of one or more parts of the text data, wherein each formulated expression can be constructed as one or more combinations of one or more phrases in a phrase thesaurus; and

means for automatically using the phrase thesaurus to construct one or more equivalent expressions of one or more formulated expressions, wherein the recognition grammar comprises the collection of all of the expressions.

19. A system as in claim 18, wherein the means for formulating an expression of each of one or more anticipated spoken inputs to the interface further comprises a graphical user interface device.

20. A system as in claim 18, wherein the means for formulating an expression representing a part of the text data further comprises means for formulating an expression representing an interactive part of the text data.

21. A system as in claim 18, wherein the means for formulating an expression representing a part of the text data further comprises means for formulating an expression representing a non-interactive part of the text data.

22. A system as in claim 18, wherein the text data represents one or more Web pages.

23. A system as in claim 22, wherein the means for formulating an expression further comprises means for automatically parsing code representing the one or more Web pages to identify the one or more parts.

24. A system as in claim 23, wherein one or more of the parts of the one or more Web pages comprise a system prompt indicating a type of interaction with an interactive part of a web page and a plurality of option values each representing a possible input to the interactive user interface for that type of interaction, and wherein the means for parsing further comprises means for identifying the system prompt and the plurality of option values.

25. A system as in claim 24, wherein the means for formulating an expression further comprises:

means for automatically identifying one or more phrases that correspond to a system prompt; and

means for automatically identifying, for each of a plurality of option values, one or more phrases that correspond to the option value.

26. A system as in claim 22, wherein the means for automatically parsing further comprises means for automatically parsing code expressed in a markup language.

27. A system as in claim 26, wherein the means for automatically parsing further comprises means for automatically parsing code expressed in HTML.

28. A system as in claim 26, wherein the means for automatically parsing further comprises means for automatically parsing code expressed in XML.

29. A system as in claim 22, wherein the recognition grammar is expressed in VXML.

30. A system as in claim 18, further comprising data storage means for storing data representing the phrase thesaurus and the recognition grammar.

31. A system as in claim 30, wherein the data storage means further stores data representing lexical items that can be used to complete a phrase template.

32. A system as in claim 30, wherein the data storage means further stores data representing a probability of occurrence of phrases.

33. A system as in claim 32, wherein the phrases are stored in the data storage means in accordance with the corresponding probability of occurrence.

34. A system as in claim 32, wherein the means for automatically using the phrase thesaurus to construct equivalent phrases further comprises means for using the data representing a probability of occurrence of phrases to construct a probabilistic grammar.

35. A system as in claim 1, wherein the means for automatically using the phrase thesaurus to construct equivalent phrases further comprises:

means for selecting a combination of one or more phrases representing a formulated expression, wherein the phrases of the selected combination of one or more phrases are original phrases of the formulated expression;

means for identifying an equivalent phrase for each of one or more original phrases of the formulated expression; and

means for producing a new combination of one or more phrases representing the formulated expression, the new combination including at least one of the identified equivalent phrases, wherein the new combination represents the equivalent expression.

36. A system as in claim 25, wherein:

phrases in the phrase thesaurus have a probability of occurrence associated therewith;

one or more original phrases has a plurality of equivalent phrases; and

the means for identifying an equivalent phrase further comprises means for selecting an equivalent phrase having the highest probability of occurrence.

37. A system as in claim 35, wherein equivalent phrases are grouped in classes and each class of equivalent phrases has associated therewith a descriptor denoting a conceptual representation of the phrases contained in that phrase class, the system further comprising means for tagging each equivalent expression with the descriptor or descriptors associated with phrases of the equivalent expression.

38. A system as in claim 18, further comprising means for translating the recognition grammar into a form that can be processed by a speech recognition system.

39. A system as in claim 18, further comprising means for manually editing the recognition grammar.

40. A system as in claim 18, wherein expressions representing a plurality of parts of the text data are formulated and the phrase thesaurus is used to identify equivalent expressions for a plurality of formulated expressions.

41. A computer readable storage medium encoded with one or more computer programs for creating a recognition grammar for use with an interactive user interface to human readable text data that is also machine readable, the interactive user interface being responsive to spoken input, the computer programs comprising:

instructions for formulating an expression representing a part of the text data for each of one or more parts of the text data, wherein each formulated expression can be constructed as one or more combinations of one or more phrases in a phrase thesaurus; and

instructions for automatically using the phrase thesaurus to construct one or more equivalent expressions of one or more formulated expressions, wherein the recognition grammar comprises the collection of all of the expressions.

42. A computer readable storage medium as in claim 41, wherein the instructions for formulating an expression representing a part of the text data further comprise instructions for formulating an expression representing an interactive part of the text data.

43. A computer readable storage medium as in claim 41, wherein the instructions for formulating an expression representing a part of the text data further comprise instructions for formulating an expression representing a non-interactive part of the text data.

44. A computer readable storage medium as in claim 41, wherein the text data represents one or more Web pages.

45. A computer readable storage medium as in claim 44, wherein the instructions for formulating an expression further comprise instructions for automatically parsing code representing the one or more Web pages to identify the one or more parts.

46. A computer readable storage medium as in claim 45, wherein one or more of the parts of the one or more Web pages comprise a system prompt indicating a type of interaction with an interactive part of a web page and a plurality of option values each representing a possible input to the interactive user interface for that type of interaction, the instructions for parsing further comprising instructions for identifying the system prompt and the plurality of option values.

47. A computer readable storage medium as in claim 46, wherein the instructions for formulating an expression further comprise:

instructions for automatically identifying one or more phrases that correspond to a system prompt; and

instructions for automatically identifying, for each of a plurality of option values, one or more phrases that correspond to the option value.

48. A computer readable storage medium as in claim 44, wherein the instructions for automatically parsing further comprise instructions for automatically parsing code expressed in a markup language.

49. A computer readable storage medium as in claim 48, wherein the instructions for automatically parsing further comprise instructions for automatically parsing code expressed in HTML.

50. A computer readable storage medium as in claim 48, wherein the instructions for automatically parsing further comprise instructions for automatically parsing code expressed in XML.

51. A computer readable storage medium as in claim 44, wherein the recognition grammar is expressed in VXML.

52. A computer readable storage medium as in claim 41, wherein the instructions for automatically using the phrase thesaurus to construct equivalent phrases further comprise:

instructions for selecting a combination of one or more phrases representing a formulated expression, wherein the phrases of the selected combination of one or more phrases are original phrases of the formulated expression;

instructions for identifying an equivalent phrase for each of one or more original phrases of the formulated expression; and

instructions for producing a new combination of one or more phrases representing the formulated expression, the new combination including at least one of the identified equivalent phrases, wherein the new combination represents the equivalent expression.

53. A computer readable storage medium as in claim 52, wherein:

phrases in the phrase thesaurus have a probability of occurrence associated therewith;

one or more original phrases has a plurality of equivalent phrases; and

the instructions for identifying an equivalent phrase further comprise instructions for selecting an equivalent phrase having the highest probability of occurrence.

54. A computer readable storage medium as in claim 52, wherein equivalent phrases are grouped in classes and each class of equivalent phrases has associated therewith a descriptor denoting a conceptual representation of the phrases contained in that phrase class, the one or more computer programs further comprising instructions for tagging each equivalent expression with the descriptor or descriptors associated with phrases of the equivalent expression.

55. A computer readable storage medium as in claim 41, further comprising instructions for translating the recognition grammar into a form that can be processed by a speech recognition system.

56. A computer readable storage medium as in claim 41, further comprising instructions for manually editing the recognition grammar.

57. A computer readable storage medium as in claim 41, wherein expressions representing a plurality of parts of the text data are formulated and the phrase thesaurus is used to identify equivalent expressions for a plurality of formulated expressions.

* * * * *