Applying Natural Language Processing Techniques to Augmentative Communication Systems

Kathleen McCoy, Patrick Demasco, Mark Jones, Christopher Pennington & Charles Rowe
Applied Science and Engineering Laboratories, University of Delaware, A.I. duPont Institute
P.O. Box 269, Wilmington, DE, USA19899

Introduction

A large population of non-speaking, motor-impaired individuals must rely on computer-based communication aids. These aids, in essence, present the user with a list of letters, words, or phrases that may be selected to compose the desired utterance. The resulting utterance may then be passed to a speech synthesizer or document preparation system. While many of these individuals desire to communicate in complete well-formed sentences, the expenditure in effort and time is often prohibitive. The goal of this project is to increase the communication rate of physically disabled individuals by cutting down the number of items that must be selected in order to compose a well-formed utterance. We wish to do this while placing as little a burden on the user as possible.

Input to our system are the uninflected content words of the desired utterance, consider, "APPLE EAT JOHN". The system first employs a semantic parser to form a semantic representation of the input. In this example, the parser must recognize that EAT can be a verb which accepts an animate ACTOR and an inanimate/food OBJECT. The resulting semantic representation (along with a specification of the original word order) is then passed to the translator which is responsible for replacing the semantic terms with their language-specific instantiations. The final phase of processing is a sentence generator which attempts to form a syntactically correct sentence that retains the general order of the original input producing, for example, "THE APPLE IS EATEN BY JOHN"

In this paper we discuss the three processing phases described above as well as examples illustrating the current capabilities of the system.

Semantic Parser

This sub-system is responsible for generating a set of semantic structures (based on Fillmore's case frames[Fillmore77]) representing possible interpretations of the input sentence.

Due to the severe syntactic ill-formedness of our input and the relatively unconstrained domain of discourse, our system may not rely on syntactic or domain specific cues. Instead, our parser relies (in a bottom-up fashion) on semantic information associated with individual words in order to determine how the words are contributing to the sentence as a whole [Small & Rieger 82]. In addition, we employ a top-down component which ensures that the individual words are fit together to form a well-formed semantic structure. Both processing components are driven by knowledge sources associated with the system's lexical items.

The first problem faced by the system is determining the general function of each word in the sentence. Each individual word can have different semantic classifications and thus its function in the sentence may be ambiguous. For example, the word "study" has two meanings: an action, as in "John studies", or a location, as in "John is in his study". In order to recognize all possible meanings of a word (and to constrain further processing) we employ five hierarchies of word meaning. Each hierarchy represents a different semantic function that a word can play: Actions (verbs), Objects, Descriptive Lexicons (adjectives), Modifiers to the Descriptive Lexicon (adverbs), and Prepositions. Distinctions within each hierarchy provide a finer granularity of knowledge representation. For example, distinguishing which objects are animate.

For each word of the input, a subframe is generated which indicates the word's semantic type for each of its occurrences in the hierarchies.Each individual word is likely to have a number of interpretations (i.e., subframes). However, if the input words are taken together, then many of the interpretations can be eliminated. In the case frame semantic representation we have chosen, the main verb of an utterance is instrumental in eliminating interpretations of the other input words. We employ additional knowledge associated with the items in the VERBS hierarchy to capture this predictive information.The main verb predicts which semantic roles are mandatory and which roles should never appear, as well as type information concerning possible fillers of each role. For example, the verb "go" cannot have a THEME case in the semantic structure. Furthermore, it cannot have a FROM-LOC case without having a TO-LOC at the same time. But "go" can take a TO-LOC without a FROM-LOC. Since specific types of verbs have their own predictions on the final structure, we attach predictive frames which encode a possible sentence interpretation to each verb type in the hierarchy of VERBS. The frames contain typed variables where words from the input can be fit, and act as a top-down influence on the final sentence structure. They can be used to reduce the number of interpretations of ambiguous words because they dictate types of words which must and types that cannot occur in an utterance.

A final level of ambiguity remains, however. A particular input word may not be modifying the verb (and thus fit into the verb frame), but rather may be modifying another non-verb input word.To reduce this ambiguity the system employs a knowledge structure that specifies the kind of modifiers that can be attached to various types of words. Thus for example "green" may be restricted from modifying "idea".

Given these knowledge sources the system works both top-down and bottom-up. With the initial frames for the individual input words, the system attempts to find a legal interpretation based on each possible verb found in the input. In a top-down way, the frames resulting from a particular choice of verb attempt to find words of specific types to fill their variables. Bottom-up processing considers the meaning of each individual word and the modification relationships which may hold between words. It attempts to put individual words together to form sub-frames which take on the semantic type of the word being modified. These sub-frames are eventually used to fill the frame structure obtained from top-down processing.

The result of this processing is a set of (partially filled) semantic structures. All well-formed structures (i.e., all structures whose mandatory roles have been filled and which have been able to accommodate each word of the input) are potential interpretations of the user's input and are passed one at a time to the next component of the system.[1]

Translator

The next phase, the translator, acts as the interface between the semantic parser and the generator. It takes the semantic representation of the sentence as input and associates language specific information to be passed to the generator component. Following [McDonald 80, McKeown 82] it replaces each element of the semantic structure with a specification of how that element could be realized in English. Each component type in the semantic message has an entry in the translator's "dictionary" that holds its possible structural translations. The actual translation chosen may be dependent on other semantic elements. When the "dictionary" is accessed for a particular semantic element, we give it both the element and the rest of the semantic structure. The "dictionary" returns a transformed structure containing the translation of the particular element along with annotations that may affect the eventual syntactic realization.

Generator

The final phase, the generator, uses a functional unification grammar [Kay 79] in order to generate a syntactically well-formed English sentence. We employ a functional unification grammar generator provided by Columbia University [Elhadad 88]. The fundamental unit of a functional unification grammar is the attribute-value (A-V) pair. Attributes specify syntactic, semantic, or functional categories and the values are the legal fillers for the attributes. The values may be complex (e.g., a set of A-V pairs). This type of grammar is particularly attractive for sentence generation because it allows the input to the generator to be in functional terms. Thus the language specific knowledge needed in the translation component can be minimized.

In the functional unification model, both the input and the grammar are represented in the same formalism, as sets of A-V pairs each containing "variables". The grammar specifies legal syntactic realizations of various functional units. It contains variables where the actual lexical items that specify those functional units must be fit. The input, on the other hand, contains a specification of a particular functional unit, but contains variables where the syntactic specification of this unit must occur.

The generator works by the process of unification where variables in the grammar are filled by the input and input variables are filled by the grammar. The resulting specification precisely identifies an English utterance realizing the generator input.

Current Status

An implementation of the system has been completed and is currently being evaluated.The system is a back-end system which takes as input the uninflected content words of the target sentence. Output from the system is a set of semantically and syntactically well-formed sentences which are possible interpretations of the input. Before the system can actually be deployed to the disabled community, it must be provided with a front-end system which will provide the potential words to the user for selection. In

[1] Our system does not handle metaphors

addition, the front-end must allow the user to select the intended utterance from the ones provided when the system finds more than one interpretation for the input. Care has been made in the design of the back-end system so that it will be compatible with many kinds of front-end systems being developed today.

System capabilities are illustrated below.

Input:	John call Mary
Output:	John calls Mary
Output:	John *is* call*ed by* Mary

Notice that it is unclear which of John or Mary is playing the AGENT and THEME roles since they both have the same type and that type is appropriate for both roles. In such instances of ambiguity all possible structures are output. In this particular example, the passive form was chosen in an attempt to preserve the user's original word order. Note that if the verb of the sentence could not undergo the passive transformation, only one option would be given.

Input:	John study weather university
Output	John stud*ies* weather *at the* university
Input:	John read book study
Output:	John reads *the* book *in the* study

The above set illustrates multiple meanings of some words. Even though study can be both a verb and a place, in the first instance it is taken as the verb since neither weather, university, nor John are appropriate. Notice in the second example study is taken as a place since the system cannot find a reasonable interpretation with study as the verb.

The first example of this set also illustrates the top-down processing power. Here, the system correctly infers weather to be the THEME and university to be the LOCATION. While technically university could be the THEME of study, weather is appropriate for no other role. Note the appropriate prepositions are used to introduce the roles.

In some cases, our system is capable of inferring the verb intended by the user even though it is unstated. Since our analysis indicates that the verbs HAVE and BE are often intended but left unstated, the system chooses between these verbs when the verb is left implicit by the user. The chosen verb is dependent on the suitability of the other input elements to play the mandatory roles of the verb.

The system may also infer the actor (subject) of the intended sentence. In particular, if no word that can play the role of agent is given, the system will infer the user to be the agent, and thus generate a first person pronoun for that slot.[2]

Input:	hungry
Output:	*I am* hungry
Input:	John paper
Output:	John *has the* paper

Conclusion

We have successfully applied natural language processing techniques to augmentative communication. Our problem is one of expanding compressed input in order to generate well-formed sentences. It is a real-world problem, we have not relied on limited domain or micro-world assumptions. The solution has required innovations in both understanding and generation. Our system must first understand a severely ill-formed utterance. The resulting semantic representation. is then translated into a well-formed English sentence. This process may require inferring such elements as function words verb morphology and, some content words.

References

C.J. Fillmore.The case for case reopened. In P. Cole and J.M. Sadock, editors, *Syntax and Semantics VIII Grammatical Relations*, pages 59-81, Academic Press, New York, 1977.

Elhadad, M.The FUF Functional Unifier: User's manual. Technical Report # CUCS-408-88 Columbia University, 1988.

M. Kay. Functional grammar. In *Proceedings of the 5th Annual Meeting*, Berkeley Linguistics Society, 1979.

D.D. McDonald. *Natural Language Production as a Process of Decision Making Under Constraint*. Ph.D. thesis, MIT, 1980.

K.R. McKeown. Discourse strategies for generating natural-language text. *Artificial Intelligence*, 27(1):1-41, 1985.

S. Small and C. Rieger. Parsing and comprehending with word experts (a theory and its realization). In Wendy G. Lehnert and Martin H. Ringle, Editors, *Strategies for Natural Language Processing*, 1982.

Acknowledgments

This work is supported by Grant Number H133E80015 from the National Institute on Disability and Rehabilitation Research. Additional support has been provided by the Nemours Foundation.

[2] We plan to employ an analysis of previous utterances to infer agents in a more general case.