Towards More Intelligent AAC Interfaces: The Use of Natural Language Processing

Patrick Demasco, Kathleen McCoy, Yu Gong, Christopher Pennington, Charles Rowe
Applied Science and Engineering Laboratories
University of Delaware/A.I. DuPont Institute

Abstract

This paper discusses the use of natural language processing techniques in the design of language interfaces for augmentative communication systems. It presents a new technique called sentence compansion that utilizes both natural language understanding techniques and generation techniques to allow a user to express full grammatical sentences with compressed content word sequences.

Introduction

One of the major goals of augmentative communication research has been to improve the interface between the user and the system. This interface can be viewed as consisting of two components. The physical interface transfers user motor acts into meaningful system input. The language interface provides the user with a set of linguistic units to select from via the physical interface. Improvements to this component of the interface can be obtained from a wide variety of techniques.

Background

At the simplest level, the language interface can be viewed as a structured collection of linguistic units (also called vocabulary sets). Early devices implemented vocabulary sets as simple two dimensional matrices that corresponded to a physical keyboard or to a visual keyboard (e.g., scanning). An extension to this structure that is commonly used today is the incorporation levels. A system with levels incorporates mechanisms to map different matrices to the same physical interface, thus allowing much larger vocabulary sets.

A great deal of effort has been devoted towards further improvement of the language interface. One of the earliest techniques used was optimization of vocabulary sets. Optimization is based on the nonuniform access time to specific units in the vocabulary set. For example, in a simple row column scanning array, the access time to an item is proportional to its distance from the upper left hand corner of the display. Foulds et. al. [1975] utilized this characteristic to design an optimized alphabetic vocabulary set.

Optimization is possible because the human language is structured and rule governed. The keyboard optimization discussed above relied on the fact that in general, letters occur with varying frequencies. Because ‘e’ occurs more frequently than ‘z’, it should be more accessible to the user.

Letter sequence frequency has also been exploited in other ways such as in linguistic prediction. Predictive systems rely not only on general information about language, but also on information obtained from the user. A predictive system will dynamically alter the structure of the language interface by offering the user next probable selections based on his/her previous selections [Baletsa et. al., 1976]. Prediction is distinguished from optimization by two important characteristics: 1) the system is dynamic (and in theory more responsive to the user's needs), 2) the system relies on more linguistic information.

To continue making progress in the design of better language interfaces, it is necessary to utilize techniques that take greater advantage of the structure of human language and derive more information from the device user. A branch of computer science called *natural language processing* provides some of the tools to do this.

Natural language processing (NLP) techniques can be distinguished from current methods used in AAC by two major factors. First, most techniques currently used (e.g., prediction) rely heavily on language statistics. While statistics are extremely useful, NLP's emphasis on language rules provide the potential to create more flexible rule based systems. Second, because language statistics are primarily limited to letter constructs (e.g., bigram, trigram, etc.), many of the currently available techniques are limited to word production. Current NLP research is active at all levels of linguistic complexity. Phonology and morphology address how words are formed. Syntax and semantics address how words are put together into meaningful sentences. Finally, pragmatics deals with conversational information that transcends the simple meaning of constituent sentences.

Sentence Compansion

Our first project that utilizes natural language processing techniques involves a new technique that we call *compansion*. Compansion is a term derived from the two words ***comp****ression* and *ex****pansion***. In general, the technique allows the user to input compressed sequences of linguistic units. The system will dynamically expand these units into a more "appropriate" representation. Sentence compansion works at the sentence level. The user inputs a sequence of content words, and the system produces a grammatically correct sentence that expresses the user's meaning. It currently assumes that the user's vocabulary set consist solely of words. The technique is best illustrated by an example:

`INPUT: John Mary go store yesterday`

`OUTPUT: John and Mary went to the store yesterday.`

Several transformations were performed in this example. First, several words were added to the input sentence (and, to, the). This lessens the total number of selections required from the user. Second, the verb "go" was properly conjugated in terms of plurality (i.e., subject verb agreement) and tense. This type of word modification will help in limiting the size of the user's vocabulary set. It is worthwhile to note the difference between this technique and other currently available compansion techniques. Abbreviation expansion is a technique that allows the user to input compressed letter sequences as representations of words or phrases [Vanderheiden, 1984]. Semantic Compaction is a technique that allows the user to express words and/or phrases with iconically represented semantic concepts [Baker, 1982]. Both of these techniques use statically defined sequences that must be memorized by the user. If the code is not in the system dictionary, the system will not perform any transformations. Because of the infinite number of sentences that we can produce, these techniques work best at the word or phrase fragment production level

Results

A prototype system has been developed on a Symbolics Lisp Machine using the Common Lisp dialect. It takes as input a sequence of compressed words (correctly spelled) and outputs a syntactically and semantically well formed sentence. The system is limited to input that is contained within its lexicon. The current system has three major functional components.

Semantic Parser - The semantic parser expects as input a sequence of words and builds a semantic representation based purely on semantic information. This is different from most natural language understanding systems in that input is a sequence of content words rather than a grammatically well formed sentence.

Representation Translator- The representation translator is the interface between the semantic parser and the generator. It takes as input the semantic representation of the sentence and outputs the deep the deep structure of the sentence. It uses (among other things) some syntactic features of the input such as word order.

Generator - The generator takes as input the sentence deep structure and outputs a- grammatical sentence. It adds function words (e.g., article, prepositions) and correctly modifies root words (e.g., verbs).

At this point, the system is limited in several ways. First, since it relies heavily on the semantic features of words, we have limited the domain of its vocabulary. This is necessary during this initial phase of exploration into the technique. Second, the current implementation runs on a specialized Lisp machine, and we expect that it will be some time before the work is transferred to microcomputer hardware. Since we are in the research stage, we choose not to limit ourselves to widely available technology. We do however expect several practical spin-offs to emerge from this project as it progresses.

One potential advantage of this technique that we anticipate is a reduction of mental load compared to that required for the use of prediction systems. Soede [1986] discussed the problems inherent in the dynamic vocabulary sets of prediction systems. In a companion system the vocabulary set remains constant (although the output stream does change upon the completion of the compansion process). The mental load effects of this system have yet to be evaluated.

Although we see this is a general purpose technique, usable by many different populations, we are targeting non-spelling individuals as the primary beneficiaries. In systems that are based on sight word vocabulary sets, we would be able to reduce the total word count as well as save the user from making extra selections We also think that this approach could be of benefit to symbol-based systems. Many of the words that are difficult to represent symbolically such as verb conjugations (e.g., are, is, am, was, were), and function words (e.g., the, to) would not be necessary. Finally, this technique might find application as a language training tool.

Conclusion

Sentence compansion is just an example of the application of natural language processing techniques. The tools developed in this project can potentially be applied to many other techniques (e.g., word prediction, translators). We feel that this technology will emerge as a fundamental assist in providing future communication devices with a greater degree of intelligence.

References

Baker, B. Minspeak. *Byte*. September, 1982, p. 186ff.

Baletsa, G., Foulds, R., and Crochetiere, W. Design Parameters of an Intelligent Communication Device. *Proceedings of the 29th Annual Conference on Engineering in Medicine and Biology*, 1976, p. 371.

Foulds, R., Baletsa, G. & Crochetiere, W. The Effectiveness of Language Redundancy on Non-Vocal Communication, *Devices & Systems for the Disabled*. 1975.

Soede, M. Prediction: the Dilemma of Mental Load. *Proceedings of the 10th Annual Conference on Rehabilitation Engineering*, 1986.

Vanderheiden G. A High Efficiency Flexible Keyboard Input Acceleration Technique: Speedkey. *Proceedings of the Second International Conference on Rehabilitation Engineering*, Ottawa, Ontario Canada. 1984, 353-354

Acknowledgments

This work has been supported by a Rehabilitation Engineering Center Grant from the National Institute on Disability and Rehabilitation Research (#H133E80015). Additional support has been provided by the Nemours Foundation.

Contact

Patrick Demasco
Applied Science and Engineering Laboratories
A.I. duPont Institute
P.O. Box 269
Wilmington, DE 19899