Applying Natural Language Processing Techniques to Speech Prostheses

Ann Copestake
Center for the Study of Language and Information (CSLI)
Ventura Hall, Stanford University, Stanford, CA 94305-4115
aac@csli.stanford.edu

Abstract

In this paper, we discuss the application of Natural Language Processing (NLP) techniques to improving speech prostheses for people with severe motor disabilities. Many people who are unable to speak because of physical disability utilize text-to-speech generators as prosthetic devices. However, users of speech prostheses very often have more general loss of motor control and, despite aids such as word prediction, inputting the text is slow and difficult. For typical users, current speech prostheses have output rates which are less than a tenth of the speed of normal speech. We are exploring various techniques which could improve rates, without sacrificing flexibility of content. Here we describe the statistical word prediction techniques used in a communicator developed at CSLI and some experiments on improving prediction performance. We discuss the limitations of prediction on free text, and outline work which is in progress on utilizing constrained NL generation to make more natural interactions possible.

Introduction

The Archimedes project at CSLI is concerned with developing computer-assisted communication for people with disabilities, considering both their interaction with computers and with other individuals. We are attempting to provide practical devices for immediate needs, and also to carry out basic research on communication, which will lead to future improvements in these techniques. The work described in this paper concerns the communication needs of people who have unintelligible speech or who lack speech altogether because of motor disabilities. It is possible to build prosthetic devices for such users by linking a suitable physical interface with a speech generator, such as DecTalk, so that text or other symbolic input can be converted to speech. However, while speech rates in normal conversation are around 150-200 words per minute (wpm), and reasonably skilled typists can achieve rates of 60 wpm, conditions which impair physical ability to speak usually cause more general loss of motor function and typically speech prosthesis users can only output about 10-15 wpm. This prevents natural conversation, not simply because of the time which is taken but because the delays completely disrupt the usual processes of turn-taking. Thus the other speaker finds it hard to avoid interrupting the prosthesis user.

This problem can be alleviated in two ways: by improving the design of the interface (keyboard, head stick, head pointer, eye tracker etc) or by minimizing the input that is required for a given output. We will concentrate on the latter aspect here, although there is some interdependence and we will briefly mention some aspects of this below.

Techniques which have been used for minimizing input include the following:

Abbreviations, icons and alternative languages Interfaces to speech prostheses commonly allow text to be associated with particular abbreviations, function keys or on-screen icons. This is useful for text which is repeated frequently, but requires memorization and does not allow much flexibility. Some systems use Minspeak (Baker, 1982), which allows short sequences of keystrokes to be used to produce whole utterances. Minspeak is compact and quite flexible, but using it effectively requires considerable learning.

Fixed text dialogues and stored narratives Alm et al (1992) describe an approach where fixed text is stored and can be retrieved as appropriate for particular stages of conversation, in particular: greeting, acknowledgement of interest/understanding, parting. Other work by the same group allows the retrieval of preconstructed utterances and stories in appropriate contexts (Newell et al, 1995). A commercial implementation of this work is Talk:About, produced by Don Johnson Incorporated. This approach is important in that it emphasizes the social role of conversation, but it allows the user little flexibility at the time the conversation is taking place.

Word (or phrase) prediction Many speech prostheses which take text rather than iconic input incorporate some kind of word prediction, where the user is given a choice of a number of words, which is successively refined as keystrokes are entered. We discuss prediction further below.

Compansion Compansion is an approach which has been used in a prototype system (Demasco and McCoy, 1992) but is not yet available commercially. It allows the user to input only content words, to which morphological information and function words are added by the system to produce a well-formed sentence. A rather similar approach is described by Vaillant and Checler (1995), though they assume icons as input. Compansion is useful for individuals who have difficulty with syntax, but as a technique for improving speech rate for users with no cognitive disability it has limitations, since it still requires the user to input at least 60% of the number of keystrokes which would be necessary to input the text normally. Furthermore, it involves natural language generation and requires considerable hand-coded linguistic knowledge (grammar, lexicon, some information about word meaning).

In the Archimedes project, we are concentrating in particular on the needs of individuals who have degenerative muscular disorders, such as amyotrophic lateral sclerosis (ALS or Lou Gehrig's disease). Individuals with ALS have no cognitive or linguistic impairment and have previously had full language use, so solutions to the communication problem which restrict their range of expression are not acceptable. Such users would prefer to continue using their original language, rather than to learn an alternative symbol system. Thus, of the techniques described above which are currently available, the most suitable is text prediction combined with single-key encoding of very frequently used phrases. Text input using a conventional keyboard may be possible, but is usually slow and painful. ALS is a progressive disease and in its later stages only eye movement may be possible, so it is important that any prosthetic system allows a range of physical input devices, since interfaces which are most suitable in the earlier stages of the disease will become unusable later on.

We found that, in addition to the speed problems, existing commercial text-to-speech systems which incorporated word prediction had a variety of drawbacks. In particular, most are dedicated to speech output and cannot be used to aid writing text or email. There are also problems of limitations in compatibility with particular software or hardware, and restrictions in the physical interfaces. One of the fundamental engineering principles underlying the Archimedes project is that individuals should have Personal Accessors which take care of their personal needs with respect to physical input and which can be hooked up to any host computer, with a small inexpensive adapter, replacing the conventional keyboard and mouse. A Personal Accessor for one user with ALS has been developed at CSLI, and now forms his main means of communication. Currently a keyboard is used for input, modified so that all the keys are mapped to the right hand side since the user has very restricted right hand mobility and no left hand use. An interface to an eye-tracker is currently under development at CSLI to replace the keyboard. The prediction techniques which the accessor currently utilizes are discussed below, followed by an outline of work in progress on a more complex system.

Statistical Prediction Techniques

The basic technique behind word prediction is to give the user a choice of the words (or words and phrases) which are calculated to be the most likely, based on the previous input. The choices are usually displayed as some sort of menu: if the user selects one of the items, that word is output, but if no appropriate choice is present, the user continues entering letters. For example, if 't' has been entered as the first letter of a word, the system might show a menu containing *the*, *to*, *that*, *they*, *too*, *turn*, *telephone* and *thank you*. If none of these are correct and the user enters 'a', the options might change to *take*, *table* and so on. If *table* is then selected, it will be output (possibly with a trailing space) and some effort has been saved. This approach is very flexible, since the user can input anything: the worst case is that the string is unknown, in which case the system will not make any useful prediction. Prediction ordering is based on the user's previous input, and so the system can automatically adapt to the individual user. Unknown strings are added to the database, thus allowing them to be predicted subsequently.

Prediction systems have been used for at least 20 years (see Newell et al, 1995, and references therein). The basic techniques are actually useful for any sequence of actions: for example, they can be used for computer commands (e.g. Darragh and Witten, 1992). However, we will concentrate on text input here, since it is possible to improve prediction rates by using knowledge of language. For text input, the simplest technique is to use the initial letters of a word as context, and to predict words on the basis of their frequencies. The prediction database is thus simply a wordlist with associated frequencies. This basic approach was implemented in the first version of the CSLI personal accessor, using a wordlist containing about 3000 words extracted from 26,000 words of collected data as starting data.

We also built a testbed system in order to simulate the effects of various algorithms on the collected data in advance of trying them out with a user. We used a testing methodology where the data was split into training and test sets, with the test set (10% of total data) treated as unseen. We used the following scoring method:

$$\frac{(\text{keystrokes} + \text{menu selections}) * 100}{\text{keystrokes needed without prediction}}$$

(we use 'keystroke' to refer to any atomic user input, irrespective of whether the physical interface is actually a keyboard or not). For example, choosing *table* after inputting ‘t’, ‘a’ would give a score of $(2+1)/6$ or 50%, assuming that a space was automatically output after the word. This scoring method is an idealization, since it ignores the difference between positions on the menu, and assumes that the user always chooses a menu item when one is available. These assumptions are reasonable for the current system, since the choice between menu items is made using a single key, and, for a user with severe motor impairment, the cognitive effort of looking at the menu choices is small compared to the physical effort of moving the hand to a different key. With a menu size of 8, the mean score for the basic method was 57.3%.

We describe some improvements to the basic approach in the remainder of this section, since although these do not represent any major breakthroughs in prediction technology, we hope that the discussion will make the nature of the problem clearer. The techniques described were all ones that could be implemented quickly, utilizing public domain or readily available resources.

Recency

This refers to increasing the weights of recently seen words so that these are temporarily preferred. We tried a range of different factors and strategies for allowing the added weights to decay. For example, it seemed plausible that the weights of very frequent closed class words, such as *the* and *in* shouldn't be modified, while open class nouns and verbs should be. However we got the best results from the simplest strategy of adding a constant amount to the weight of a word, which was calculated to be sufficient to raise any word to the menu shown after an initial letter has been input. (Words which are already on this menu may be raised to the top level menu.) E.g. if *table* is usually predicted after seeing ‘t’ ‘a’, the recency factor was calculated to be sufficient to promote it to the menu seen after ‘t’ alone is input. The weights are removed when there is a gap of more than 10 minutes between inputs, which is a heuristic that suggests a particular conversation has ended. This simple technique results in an improvement in score from 57.3% to 56.0%.

Unknown words

In a representative test corpus of 2600 words we found that there were 193 word types which were unseen in the 23,400 word training corpus. Of these, about a third were typographical errors or proper names. A larger wordlist could thus potentially improve performance. We tried adding an extra 18,000 words extracted from a newspaper corpus to the lexicon, giving them a frequency of 0, so they would only be shown on the menus when there was no possible previously seen word. This covered 92 of the missing words. Adding it improved performance, but only by about 0.9%, and at the cost of greatly increasing the memory taken by the system and decreasing its speed. A more targeted approach is to add inflectional variants of words which have been seen. E.g. if *communicate* is seen, the system can also add *communicates*, *communicated* and *communicating*. This technique results in 41 of the unseen words being added, and has the advantage that fewer inappropriate words are shown to the user. (The freely-available morphology system described in Karp et al (1992) was used for this work.) This technique is most useful when used in conjunction with the syntactic filtering technique described below, since this also requires morphological analysis in order to determine syntactic category.

ngrams

Given the great improvements that have been made in speech recognition by using Hidden Markov Models (HMMs), it is natural to expect that these techniques would be beneficial for word prediction. However existing text corpora do not make good models for the speech of our user, and the amount of training data which we can collect is insufficient to extract reliable word-based trigrams. 26,000 words of data represents around three months of input, and much more data would be necessary to collect useful trigrams. One solution to this problem is to back off to broader categories, so we investigated the use of part of speech (POS) bigrams extracted from existing tagged corpora available from the Linguistic Data Consortium (Penn Treebank). The idea is that we can use a corpus in which each word token has been tagged with its appropriate POS in that context and derive transition probabilities for POS-to-POS transitions, instead of word-to-word transitions. Then, if the predictor modifies the weights on words according to their possible parts-of-speech and the possible parts-of-speech of the previous word, we get some syntactic filtering. For example, if word frequencies alone are used, the top eight possibilities predicted by one of our training sets when no initial letter has been input were *I*, *is*, *the*, *you*, *it*, *to*, *a* and *ok*. These are all almost certainly inappropriate following *we* at the start of the sentence. This is partially reflected in POS transition probabilities: e.g. the probability of PP (personal pronoun) being followed by another PP is very low. Using transition probabilities extracted from a subset of the tagged Treebank corpora to modify the frequencies resulted in the following list being predicted following *we*: *is*, *did*, *are*, *was*, *ok*, *call*, *should* and *can*. Here five of the eight menu items are reasonably probable continuations, thus we have achieved some syntactic filtering. The inappropriate words *is* and *was* are due to *we* being tagged simply as PP since the Treebank tagset we used does not distinguish between singular and plurals pronouns. *OK* is predicted because it was unseen in the corpus data, and so did not have an allocated tag: words without a tag were treated as though they could belong to any class for this initial experiment.

However, overall we got no improvement in performance using POS transition probabilities extracted from the Treebank corpora, apparently because they were a poor model for our data. The problem seems to be that our user makes much more frequent use of questions, imperatives and interjections than the sentences found in the corpora we used to extract the POS bigrams. We therefore decided to derive transition probabilities directly from the data collected from our user. As an initial step we tried to tag our data using the tagger developed by Elworthy (1994) and a lexicon derived from the Treebank corpora. This gave some unexpected results: the Treebank corpus turned out to be a good model for our data with respect to relative frequencies of POS associated with particular words (e.g. if a word such as *table* is likely to be a noun rather than a verb in the Treebank corpus, then on the whole the same is true in our user's data). Because of this, we got about 92% tagging accuracy simply by choosing the most frequent tag for each word. Running the tagger did not improve these results, which is unsurprising since taggers generally perform rather poorly when the initial data is close to being correct, where the dataset is small, and when the sentences are short, all of which were true here. Using the POS transition probabilities derived from a text which was tagged by simply taking the most globally probable tags for the word in the training data gave an overall improvement in prediction rate of about 2.7%. We expect to be able to improve this somewhat, using a refined set of POS tags and considering POS trigrams instead of bigrams.

Our tentative conclusions from this work are that it is possible to construct POS bigrams and trigrams from data collected from a personal accessor, without using a tagger, by simply assuming each word in the input has its most likely POS. This will introduce some errors, but unless these are systematic, they should not affect the overall accuracy of the model too badly. Since no tagger is involved, the transitions can be straightforwardly learned by a running accessor. An external lexicon which gives the possible parts of speech for a given word and identifies the most frequent is needed, but this can be derived from existing tagged corpora. Manual augmentation of the lexicon is useful for cases where the user's vocabulary contains words which are not found in the external corpus, but this is not essential.

This approach to syntactic filtering has several advantages over using a conventional grammar. Grammars are relatively difficult to develop and maintain, and parsing is expensive computationally. If the grammar is narrow in coverage, it will not provide any predictions for sentences which are not covered. On the other hand, a broad coverage grammar would need to be augmented with probabilities in order to distinguish between likely and unlikely strings. For example, *the* can follow *we* at the beginning of a sentence, but only in rather unlikely contexts such as *we, the undersigned.* Of course the grammaticality of a string can depend on more than one or two words of previous context, but without some reliable methods for lexical and structural disambiguation, a conventional grammar will posit too much ambiguity to make very tight predictions possible. Many of our user's utterances are short phrases rather than complete sentences, so any grammar would have to be capable of dealing with fragments, but these are allowed for naturally with the POS ngram model.

Some conclusions on prediction

All the experiments reported above kept the size of the menu at eight items. Increasing the menu size improves performance, however there are obviously limitations to the menu size that a user can cope with. Eight items is at the upper end of the range usually suggested in HCI studies, but the system currently in use presents a choice of 10 items, since we found that the limiting factor was not the cognitive load of scanning that number of menu choices, but the size of the screen. A smaller menu size might be better for a user with more mobility. We hope that when using the eye-tracker it will be possible to make use of multiple menus, arranged according to content, since a wider range of options can be made available with a single movement. For example, it would seem natural to have a separate menu of proper names.

Similarly, results would improve if it was possible to predict phrases rather than words. But mixing words and phrases in a single menu is confusing (except for cases such as compound nouns, like *desk lamp*, which are in many respects wordlike in their behavior). In any case, the current system has a number of hardwired phrases tied to particular keys or accessed through a hierarchical menu, which take care of the most frequent phrases.

One disadvantage of concentrating too much on modeling the collected data is that it is far from the output that our user would like to produce. We have been collecting data for about nine months, and over that time there have been significant changes in the output: the utterances now tend to be shorter, though more frequent, and there is an increasing tendency to produce ungrammatical output, by dropping *the* for example. Furthermore, although we have concentrated on speech output here, email and other written forms have rather different characteristics, so for optimum performance different modes are necessary. Obviously we would also like to check performance with more than one user, although, since we have not done any hand-modeling in the work reported here, the system could be made to adapt to a new user incrementally and automatically.

Our current user finds word prediction is a great benefit and he will not use any system that does not incorporate it. Work on improving the prediction techniques is still in progress, and we expect to get some further performance gains. Savings in keystrokes of up to 60% have been reported in the literature, but these results appear to be from subjects who were using more restricted vocabularies. However, we feel that the experiments described here suggest that we are seeing diminishing returns from considerable increases in complexity to the system. We believe that word prediction alone cannot achieve the improvements in output rate which are needed for more natural speech. Because of this, we are in the process of developing a new technique which combines prediction with some elements of the other approaches listed in the introduction. This 'cogeneration' approach is described in the next section.

Cogeneration

Cogeneration is a novel approach to natural language generation which is under development at CSLI. Traditionally, generation has been seen as the inverse of parsing, and the input is some sort of meaning representation, such as predicate calculus expressions. This is inappropriate for assistive communication, since formal meaning representation languages are hard to learn and anyway tend to be more verbose than their natural language counterparts. Instead, in cogeneration, input is partially specified as a series of text units by the user, and the job of the generator is to combine these units into grammatical, coherent sentences which are idiomatic, appropriately polite and so on. To accomplish this, the generator has to be able to order the text units, add inflections and insert extra words (both function and content words). This concept of lexicalist generation is closely related to Shake'n Bake generation (Whitelock, 1992; Poznanski et al, 1995). The knowledge sources which are needed for cogeneration are:

- a grammar and lexicon expression in a constraint-based framework, such as Head-driven Phrase Structure Grammar (HPSG: Pollard and Sag, 1987, 1994)
- statistical information about collocations and preferred syntactic structures
- application- and context-dependent templates which are used to guide both the user input and the process of generation, and to provide fixed text for conventional situations

Thus cogeneration involves a combination of grammatical, statistical, and template-based constraints. For application to speech prostheses the templates will be designed for particular dialogue situations. These will be organized in a hierarchy which will contain general classes, such as 'request', 'question', 'statement', with more specific templates such as 'refer to earlier discussion' inheriting from the general classes. The templates provide constraints on generation, which can be expressed in the same formalism as the grammar and lexicon. The use of partial text supplied by templates combined with free user input extends the work of Alm et al (1992), which concentrated on almost completely fixed text for sequences such as greetings, to the much wider range of situations where partially predefined strings are appropriate.

The choice of template is made by the user, and the interface provides slots which the user instantiates with text units. In many cases, slots will be optional or have default fillers, constructed according to context and previous inputs. Instantiation of the slots is aided by word and phrase prediction, conditioned on slot choice. Prediction should be much more effective than with free text, since the slots will provide fine-grained syntactic and semantic constraints. The cogenerator operates by combining the constraints specified by the template(s) with the general constraints of the HPSG grammar to produce an output sentence, guided by statistical information.

To give a concrete example, the 'refer to earlier discussion' template might have slots for 'topic of discussion', 'time of discussion' (optional) and 'participants in discussion' (defaulting to 'us'). The user might input *buy desk lamp* to the topic slot, and *breakfast* in the time slot. The template would have a number of partially fixed text units associated with it, which might include:

> You know <participants> talking about <topic>

The system could then generate:

> You know we were talking at breakfast about buying a desk lamp

The intention is that the system provides the inflectional information, and uses syntactic information to arrange the text so that the sentence sounds reasonably natural. Cogeneration thus also builds on Demasco and McCoy's (1992) work on compansion. The placement of the prepositional phrase *at breakfast* after *talking* rather than at the end of the sentence is motivated because this avoids ambiguity. The expansion of *breakfast* into *at breakfast* involves the choice of a particular preposition, and choice of the indefinite article (*a desk lamp* rather than *the desk lamp*). This latter choice would be based on the (default) assumption that the particular objects bought will not have been previously mentioned, possibly reinforced by information from previous input. In general, the system will maintain a complete record of utterances, so that text can be appropriately indexed and retrieved.

The need for linguistic processing as an essential component of this approach is shown by the ungrammaticality and near-unintelligibility of the output which would have resulted from treating the user input and the template text as fixed strings:

> You know us talking about buy desk lamp breakfast

In general, fixed string substitution has severe limitations in anything other than the most restricted circumstances.

In principle there need be no restriction on user input to the system. The system would perform optimally if there were a full lexical entry for every word, but would degrade gracefully if little or no information were known about some of the words in the user input. However, full lexical entries will be needed for words in the template strings. We envisage a range of about 20 templates being sufficient to cover a wide range of conversational situations, so the cognitive load involved in learning and selecting templates will not be great. More specific templates could be customized for an individual speaker, and since the information associated with specific templates will be expressed as a text string, this could be done by the user. Similarly, the user could provide customized fixed text options for the built-in templates. The system will be inherently robust, in that if there were no appropriate specific template, the very general templates could be used as a fall-back: the user would have to type more, but the word prediction would still operate. Furthermore, the system will be able to adapt automatically to an individual user over time, both with respect to word prediction and template preferences.

Besides the hierarchical arrangement of templates, there will also be a linear ordering, which allows the user's choice of template to be predicted by the system under some circumstances. This is most apparent in highly conventional situations such as greetings. Naturally, since the system only has access to one side of the conversation, the prediction cannot be perfect. However, we would expect the user to be able to take advantage of sequences and use them to channel the conversation so that communication is most easy for them, possibly with the aid of templates which indicated 'return to previous topic' etc. Furthermore we would expect there to be environments where it is possible to provide scripts for interactions (e.g. Schank and Abelson, 1977), such as restaurants and supermarkets. Constructing detailed scripts for real world situations is not a main focus of our research, since we would expect there to be wide variability in what a particular individual would find useful. It should be possible to make the template construction tools sufficiently easy to use to allow templates and scripts to be added by the user for his/her particular needs.

Conclusion

Prediction techniques are robust and flexible, but by themselves cannot offer the improvement in text input speed necessary to allow natural conversation using a text-to-speech system. We therefore propose to use the cogeneration technique for this application. Natural language generation as usually conceived involves many complex AI issues: real world knowledge representation and reasoning, planning, user-modeling, reasoning about user goals and intentions and so on. For this reason, it is currently feasible only for systems which operate with very restricted domains and which have rather limited and well-defined communication goals. The cogeneration approach avoids any explicit encoding of real-world knowledge by being user-driven and by the combination of templates, grammar (including lexical semantics) and statistical information. We therefore expect to build a usable prototype which is capable of operating in any domain, unlimited by subject matter. The other most important feature of cogeneration compared to conventional generation is that it does not require production of a complete meaning representation, but is essentially word-based, making it possible to freely mix (semi-)fixed phrases with user input text. The user does not have to learn a new language (logical or iconic) to drive the generator but is always in control of the content of the output: the generation process is user-driven via template and word choice, and the user can always choose to reject or modify the output text. This approach to generation is thus particularly suitable for an assistive device, especially for users who have lost their previous ability to speak.

Despite the avoidance of coding real-world knowledge, cogeneration is relatively knowledge-intensive compared with simple word prediction. We hope that this will pay off in terms of improvement in conversation naturalness, although this work is only in its very early stages, and we are a long way from being able to demonstrate this. However, one potential side-effect of utilizing more linguistic knowledge is that this can also be harnessed to make generated speech more natural with respect to intonation and the expression of emotion, for example. Another potential benefit is that cogeneration could be made to accept non-textual input and produce speech output. For example, the use of an eye-tracker could make it possible to input a representation of movement more directly, by scanning from a start point to an end point on a scene representing the user's room, for example. If the start-point and end-point are labeled (e.g. as representing the user and the telephone) this input can be treated as equivalent to text, and a natural language utterance such as *Bring me the telephone* could be generated. Work is also in progress at CSLI on utilizing language constructs borrowed from ASL to enhance such use of non-textual input.

Acknowledgements

I am grateful to David Elworthy for advice on tagging, to members of the Archimedes group, and especially to Greg Edwards, who implemented the Personal Accessor described here.

References

Alm, Norman, John L. Arnott and Alan F. Newell (1992) 'Prediction and conversational momentum in an augmentative communication system', *Communications of the ACM,* **35(5)**, 47–57.

Baker, B. R. (1982) 'Minspeak', *Byte,* **7(9)**, 186–202.

Darragh, J. J. and I. H. Witten (1992) *The reactive keyboard,* Cambridge University Press.

Demasco, Patrick W. and Kathleen F. McCoy (1992) 'Generating text from compressed input:an intelligent interface for people with severe motor impediments', *Communications of the ACM,* **35(5)**, 68–78.

Elworthy, David (1994) 'Part of speech tagging and phrasal tagging', ACQUILEX 2 Working paper no 10, University of Cambridge Computer Laboratory `http://www.cl.cam.ac.uk/Research/NL/acquilex/acq2wps.html`.

Karp, Daniel, Yves Schabes, Martin Zaidel and Dania Egedi (1992) 'A freely available wide coverage morphological analyzer for English', *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92),* Nantes, France.

Newell, Alan F., John L. Arnott, Alistair Y. Cairns, Ian W. Ricketts and Peter Gregor (1995) 'Intelligent systems for speech and language impaired people: a portfolio of research' in Edwards, Alistair D. N. (ed.), *Extra-Ordinary Human-Computer Interaction,* Cambridge University Press, pp. 83–101.

Pollard, Carl and Ivan Sag (1987) *An information-based approach to syntax and semantics: Volume 1 fundamentals,* CSLI Lecture Notes 13, Stanford CA.

Pollard, Carl and Ivan Sag (1994) *Head-Driven Phrase Structure Grammar,* Chicago University Press, Chicago and CSLI Publications, Stanford.

Poznanski, Victor, John L. Beaven and Pete Whitelock (1995) 'An efficient generation algorithm for lexicalist MT', *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL-95),* Cambridge, Mass..

Schank, Roger C. and Robert P. Abelson (1977) *Scripts, plans, goals and understanding: an inquiry into human knowledge,* Lawrence Erlbaum Associates, Hillsdale, N.J..

Vaillant, Pascal and Michaël Checler (1995) 'Intelligent voice prosthesis: converting icons into natural language sentences', Computation and Language E-Print Archive:
`http://xxx.lanl.gov/abs/cmp-lg/9506018`.

Whitelock, Pete (1992) 'Shake-and-bake translation', *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92),* Nantes, France.