0743-4618/01/1704-0255 S3.00/0; Volume 17, December 2001
AAC Augmentative and Alternative Communication

Improving Word Prediction Using Markov Models and Heuristic Methods

Sheri Hunnicutt and Johan Carlberger
Department of Speech, Music and Hearing (S.H.), Department of Numerical Analysis and Computing Science (J.C.), Kungliga Tekniska Högskolan, Stockholm, Sweden

The goal of this project was to design and implement a new word predictor for Swedish that would suggest words that are more grammatically appropriate, thus presenting a lower cognitive load for users and saving significantly more keystrokes than the previous predictor. The new predictor that was designed and developed uses a probabilistic language model based on the well-established ideas of the trigram predictor for speech recognition, developed by IBM. In tests, this program has been shown to result in keystroke savings of 46% given five predictions—a substantial saving compared with the 35% savings achieved with the previous predictor.

KEY WORDS: lexicon, linguistic prediction, technology, word and class n-grams, word prediction, writing aids

A number of word prediction systems that are commonly used by individuals requiring augmentative and alternative communication (AAC) include both syntactic information with their related algorithms and various heuristic methods, such as recency promotion and word learning, in an attempt to both reduce keystrokes and make predictions more efficient to the user. It has been suggested that word predictors with syntactic and heuristic knowledge could make more accurate and appropriate predictions; that these predictions might, as a result, present a lower cognitive load for the user; and that users with dyslexia might also be steered away from confusing predictions by more appropriate predictions (Tyvand & Demasco, 1993). The motivation to provide such benefits has often inspired researchers in this area despite disappointingly small increases in keystroke savings. This article describes a system in which probabilistic models of word sequences and word class sequences are integrated with complementary heuristic methods to improve both the quality of words predicted and the keystroke savings. A review of research in syntactic and heuristic methods in word prediction and a discussion of possible language dependence precedes a description of the study.

PREVIOUS RESEARCH

Syntax

Syntactic information that has been accessed to improve word prediction has included statistics for word class sequences and rules for grammatically correct sentence structure in a number of languages. Algorithms include various types of parsers and probabilistic methods, such as Markov models and artificial neural networks (ANNs). At the present time, a few commercial systems, including Aurora and Co:Writer®, use some type of grammatical information (Boekestein, 1996). Multiword prediction (usually word pairs) is also provided in a number of systems (e.g., Aurora 3.0 for Windows,[1] Co:Writer®,[2] EZ Keys™,[3] Finish Line,[4] KeyCache[5]).

To include syntactic information in an early version of the Swedish word prediction system, a 10,000-word lexicon was marked with word class information. These word classes were typically used when studying the grammar of a language: for Swedish, these included noun, verb, adjective, and function word classes, as well as subclasses such as singular/plural, gender, definite/indefinite for nouns and adjectives, and tense for verbs. A study was first done to determine the maximal possible savings in keystrokes if word class were known. In a 1,331-word sample from a somewhat telegraphic personal communicator text, it was found that

[1]Aurora Systems, Inc., Box 43005, 4739 Willingdon Ave., Burnaby, BC V56 3H0, Canada.
[2]Don Johnston Inc., 26799 West Commerce Dr., Volo, IL 60073, USA.
[3]Words+ Inc., 1220 West Avenue J, Lancaster, CA 93534-6523, USA.
[4]Innovative Designs, Inc.
[5]OMS Development, 1921 Highland Ave., Willmette, IL 60091, USA.

perfect knowledge of the part of speech would have improved keystroke savings by 2.6% for one prediction or by 5.1% for six predictions (Hunnicutt, 1989). In a similar test for English by Swiffin, Arnott, Pickering, and Newell (1987b) at the University of Dundee, a simulation with known word class (of a total of 10 word classes) indicated a possible keystroke savings of 9% to 15%. However, using a first-order word class transition matrix and multiplying word frequency by the probability of a word, given its word class, to produce an effective word frequency in their algorithm only yielded a savings of between 0.5% and 2.0% (Swiffin, Arnott, & Newell, 1987a) in test texts in which each word was labelled with its most common word class. The texts for this study were three samples of 3,500 words each, covering three different subject areas. Prediction lists of 1, 5, and 10 words covering these subject areas were employed. The lexicon contained precisely the words in the texts, marked for word class, a similar type of word class information as used in the Swedish study. A further test indicated that known word class alone would improve keystroke savings by 4.3% to 6.4% if frequency information was not available.

A precedence-type grammar was written for the Swedish system in the late 1980s, designed to rule out words with unlikely parts of speech after a word with a known part of speech (Hunnicutt, 1989). However, for single-word predictions, this grammar ruled out correct words as often as incorrect words. This study was not carried out for multiple predictions, however, since the strategy of rejecting unlikely word classes can, in the case of multiple predictions, be replaced by priority decisions in ordering predictions.

Around 1990, a parser was developed at the University of Dundee for the word prediction system PAL, and the resulting system was called Syntax-PAL (Morris, Newell, Booth, & Arnott, 1991). A complete syntactic parse of the partial sentence was attempted for the specific purpose of helping users with their syntax rather than of increasing keyboarding efficiency. Grammatical frequency statistics guided the ordering of appropriate predictions. Words determined to be ungrammatical by the parser were shown in a separate area under the grammatically appropriate words to indicate potential problems to the user.

During the early 1990s, work in using syntax statistics was carried out by Vandyke, McCoy, and Demasco (1992) at the University of Delaware and by Tyvand and Demasco (1993) at SINTEF in Oslo, Norway. At the University of Delaware, a technique was developed to allow a syntactic predictor to make rule-based determinations about which words can follow previously written words in a text. In this development, a lexicon including syntactic information and a grammar defining legal syntactic transitions was employed. The addition of transition probabilities to the grammar to increase prediction accuracy was investigated.

In analyzing all possible sequences of class trigrams (i.e., three successive word classes in a text) used in the Brown Corpus (Kucera & Francis, 1967), Tyvand and Demasco (1993) noted that 90% of the possible combinations had zero frequency. Results from a test using a simple alphabetic lookup in a 15,000-word dictionary and a prediction list of 5 words showed an increase in keystroke savings from 33% to 37% on the sample text when syntax statistics were included. However, using a prediction approach based on word frequency, keystroke savings of 47% were achieved with a 5-word prediction list. A combined probability implementation employing both syntactic information and word frequency information was proposed.

Inclusion of grammatical categories to aid in word prediction has also been investigated for Spanish (Palazuelos, Aguilera, Ricketts, Gregor, & Claypool, 1998a; Palazuelos, Aguilera, Rodrigo, & Godino, 1998b; Palazuelos-Cagigas, Godino-Llorente, & Aguilera Navarro, 1997). In the first of these studies, a word's frequency was recalculated by weighting the lexical frequency with the probability that its grammatical class would follow the preceding two grammatical classes. This probability was accessed from a class trigram matrix. When the sequence was not present, the class bigram probability was used, referencing only the previous word's class. The results in keystroke savings for a 21,000-word text and a general lexicon with words in frequency order were about 30%. With the recalculated frequency as defined above, the keystroke savings increased by about 2%. In the two later studies, other syntactic methods were investigated. In one study, a statistical chart parser was used to parse the partial sentence up to the current word, as in the work of Vandyke et al. (1992). In a 25,673-word test text, with the number of predicted words set to 7 and the maximum number of suffixes predicted set to 4, the grammar alone gave a keystroke savings of 40.2%. Including both the grammar and adaptation through learning of new words, word bigrams (two successive words in a text), and some word trigrams (three successive words in a text), as well as generation of customized lexicons, the keystroke savings rose to 52%. In the other study, the use of ANNs showed a substantial improvement in the prediction of 20 word classes over class bigrams and class trigrams with a small training text of only 2,000 words marked with their word class. In a simulation using a network trained to predict the word class of a word given the word class of the 5 preceding words, the ANN reached a word class prediction accuracy of 47.7%, compared with 42.5% for bigrams and 45.9% for trigrams for the top choice word class. Appearance of the word class among the five more probable word classes (of 20) could be achieved 85.3% of the time with the ANN, compared with 78.3% for bigrams and 78.3% for trigrams.

Potential Differences due to Language

Given that word structure differs among languages, it can be surmised that word prediction will be more or

less difficult and that accessing a lexical entry may require more or fewer keystrokes, depending on the language. There is, however, little evidence at present to show that this is the case among the Germanic and Romance languages. In a study of five European languages, Carlson, Elenius, Granström, and Hunnicutt (1985) studied the 10,000 (approximately) most frequent words in published lexica, finding that the mean word length differs by, at most, 1.6 letters. The mean word length in letters for the English vocabulary was 6.09; for Italian, 6.39; for Swedish, 6.43; for French, 6.62; and for German, 7.69. The mean word length in phonemes for the English vocabulary was 4.96; for Italian, 5.94; for Swedish, 5.94; for French, 4.20; and for German, 6.78. It can be noted that, except for French, which has comparatively fewer phonemes per letter, the relative ordering is maintained.

For English and French, the point at which a word can be uniquely specified, assuming word identification proceeds letter by letter from the word beginning, is at the fourth phoneme. The identification curves peak at this point, differing by at least 10% of the corpus from neighboring word identification points. However, for Swedish, German, and Italian, there is a cluster of three points around five phonemes, that is, on average, a word can be uniquely identified by the fourth, fifth, or sixth phoneme. Given that the ordering of relative word lengths in letters and phonemes is the same except for French, one can predict that Swedish will require approximately one more letter than English to uniquely specify a word and, therefore, one more letter to be typed before the correct prediction would occur if there is only one word in the prediction list. Since prediction lists usually contain a number of words, however, this difference would be less noticeable in practice.

In a comparison of two word prediction systems—Predice and Predictability—using five European languages (English, Italian, Spanish, French, and German), Palazuelos et al. (1998a) obtained similar results of 30% to 37% (unspecified as to language) in keystroke savings using European Union publications that were available in all five languages. Lexica were built using one publication of about 11,000 running words, and testing was accomplished on a second publication of about 2,000 running words. Five word predictions were made available and, for Spanish, five suffixes. It was noted that, using word prediction techniques specific to a language, one of the two systems could save 48.21% of possible keystrokes. Thus, the results were inconclusive: keystroke savings for the five languages are within 7% of each other, but the addition of language-specific grammatical information can make an even greater difference and may vary among the languages. A later test gave values of 33% to 35% savings for French, Italian, and German but only 21% for Spanish and a high of 41% for English (Palazuelos S, personal communication, 2001). The low percentage savings for Spanish, however, could be raised to 46% to 47% with the addition of word bigrams or trigrams or with a parser.

It is often assumed that more keystrokes will be necessary for a particular word to appear in the prediction list if a language has many inflections. These studies, however, do not bear out this hypothesis. Word frequency may be such a strong factor that words with less frequent inflections do not influence the statistics to a greater degree.

Heuristic Methods

The basic method of word prediction is to suggest one or more possible completions after a user has typed one or more initial letters of a desired word. This approach is supported by the psycholinguistic literature. It is well documented, for example, that the initial sounds and the initial letter or letters of a word are "access points." That is, a person can guess a word faster given its initial letter(s) than given its medial or final letters (Marslen-Wilson & Welsh, 1978). That initial letters are a preferable access point for message encoding in AAC devices has also been borne out in experiments with adults who use AAC (Light, Lindsay, Siegel, & Parnes, 1990).

In the word prediction systems described previously and in many other laboratory and commercial systems, the algorithms used have employed word frequency, and usually word pair frequency as well, as their basic information source. This approach is also well grounded in psycholinguistic research. It has been shown that a person's mental lexicon is, in some sense, frequency weighted (Broadbent, 1967). As an example, the more frequent a (content) word in a language is, the faster is the reaction time of a person hearing that word in classifying it as a word of the language (Bradley, 1978).

In addition to such a frequency-based algorithm, many word prediction systems have added one or more heuristics. The most common of these are new word learning and what has been referred to as "recency." Increases in keystroke and effort savings have also been sought through automatic capitalization and provision of affixes that can be selected with a single keystroke.

New Word Learning

Automatic learning of new vocabulary items has been an important and helpful feature in most word prediction systems and has been included in many such systems from the time of their first appearance in AAC devices. In most systems, words that are typed by the user and are as yet unknown to the prediction system (i.e., words that do not already appear in the system's lexicon) are stored in a special file. These words are then available to be predicted during future text composition.

The MicroDec II used a learning element of about 148 words (a 1,500-byte block), keeping track of the

most recent session in which words were used and how frequently they were used (Heckathorne, Leibowitz, & Strysik, 1983). This was based on the results of work by Gibler (see, for example, Gibler & Childress, 1983) in which a two-component lexicon was constructed with a 200-word learning element, which could contain words of up to 16 letters in length, words in the main lexicon having been truncated to a maximum of 10 letters to save space. The list was ordered chronologically, with older words deleted as new words were added. In the MicroDec II implementation, frequency of use was also taken into account so that the oldest, least frequently used words were automatically deleted. In a later system, the PACA, the observation that vocabularies from different sources vary, even in the most commonly used words, led to the provision of an entirely learned vocabulary of 750 words with an initial vocabulary of 500 frequent words for initial predictions (Heckathorne, Voda, & Leibowitz, 1987). In the PAL system and the early Swedish systems, new words could either be added to the main lexicon or retained in several context-oriented files. PAL accommodated a maximum of about 1,000 words in such files (Swiffin, Pickering, Arnott, & Newell, 1985); the Swedish system allowed for 200 words. The MicroDEC II and PAL included only new words. In the Swedish system, new words were accompanied by recently used, less frequent words (see below). Automatic learning of new vocabulary items is standard in most laboratory and commercial word predictions systems of today, including Aurora, Co:Writer, EZ Keys, Finish Line, KeyCache, PAL (Predictability),[6] SofType™,[7] WiVik® (Boekestein, 1996),[8] and Predice.

Recency Promotion

A feature of natural language is that an occurrence of a word increases the probability of that word occurring soon again in the same text. Many word predictors take advantage of this fact to "promote" the frequency value of a word that has recently occurred in the current text, which may result in its being shown sooner in the prediction list. The decision as to what rank in the prediction list the word should be given and what text length constitutes "recently" is not algorithmic but heuristic (i.e., the decision is made based on logic and experience).

In the first author's work, the term "recency" has been borrowed from the field of cognitive psychology in which experiments in serial free recall of items in a list have led to a description of the phenomenon by a U-shaped response-accuracy curve, showing that recent items in a list are readily recalled (recency effect), the same being true to a lesser extent for early items in the list (primacy effect) but not for items in the middle of the list (Murdock, 1962). This literature may also have influenced the use of the word for other researchers. Recency promotion has been mentioned in word prediction work by many authors since word prediction got its start with AAC users in the 1980s; in fact, the use of the term "recency" has been explicit in the literature for many years (e.g., Hunnicutt, 1987; Swiffin et al., 1985) and seems to have become a recognized standard term in describing word prediction algorithms (e.g., Higginbotham, 1992; Lesher et al., 1998; Venkatagiri, 1994). Many algorithms include recency in their calculations, including those used by Aurora, Co:Writer, EZ Keys, Finish Line, HandiWORD,[9] PAL (Boekestein, 1996), and Predice.

There is little mention in the literature about how recency information is used to promote recently used vocabulary items. In the early version of the Swedish word prediction system, recency promotion was achieved by inserting the list of recently used words (the subject lexicon) into the list of frequency-ordered words at rank 200, allowing most function words in the main lexicon more priority but giving the recently used words a higher rank than other content words (Hunnicutt, 1986). It is reported that in the PAL system, recency was accounted for by giving a new word the "maximum" value of recency, but it was unclear what this value was (Swiffin et al., 1987b). A later description states that "recency takes precedence over frequency" and that the balance between recency and frequency may significantly affect performance for long-term users (Newell et al., 1992). This may indicate that this balance point was variable.

Automatic Capitalization

It appears that most research and commercial word prediction algorithms automatically capitalize the first letter of the initial word in a sentence, that is to say the first letter of the first word in a text or the first letter after a terminal punctuation mark (for reviews of commercial systems, see Higginbotham, 1992, and Boekestein, 1996). In fact, this feature is rarely mentioned in the research literature, perhaps because it is such an obvious one to implement. However, it does save one keystroke for each sentence. Neither is it mentioned that words (generally proper nouns) appearing in a prediction list with an initial capital are entered in text with the appropriate initial capital without any effort on the user's part, again saving a keystroke.

[6]Inclusive Technology Ltd., Gatehead Business Park, Delph New Road, Delph, Oldham 0L3 5BX, UK.

[7]Origin Instruments, 854 Greenview Drive, Grand Prairie, TX 75050, USA.

[8]Prentke Romich Company, 1022 Heyl Rd., Wooster, OH 44691, USA.

[9]Microsystems Software, Inc., 600 Worcester Road, Framingham, MA 01702, USA.

Providing Inflected Forms of Words

Special access to prefixes and suffixes is another known heuristic feature in word prediction systems, but, as with capitalization, it is rarely mentioned in the research literature. The use of abbreviations or special affix windows are, however, reported to give large keystroke savings in commercial English-language systems (Higginbotham, 1992). Suffix lists are provided in a number of systems, including Aurora, Co:Writer, EZ Keys, HandiWORD, SofType (Boekestein, 1996), and Predice. For a language with substantially more inflections than English, these strategies are cumbersome. In the Swedish system, a strategy was introduced in which the main lexicon contained only base forms for content words, each one being marked by its inflection category (Hunnicutt, Bertenstam, & Raghavendra, 1994). When such a base form was chosen in the prediction list, the algorithm chose the appropriate set of inflections and base form adjustments, constructing a list of inflected word forms. The grammatical forms in the list were always in the same order so that, for example, the simple past tense of a verb was always in the third position and was always chosen with the third function key, F3. A short list was also made available in which only the four most common inflected forms were provided.

THE SWEDISH PROBABILISTIC MODEL

The Predict (later Prophet) word prediction program for Swedish was developed in the early 1980s at the Department of Speech, Music and Hearing at the Kungliga Tekniska Högskolan (Hunnicutt, 1986). Revisions of the program have been made continuously since this time. The program has also been localized into a number of European languages. In a Swedish national project that, unfortunately, did not come to completion, a newer probabilistic version of this word prediction program was developed for Swedish. It was to be a component of a program developed by several groups in Sweden to be used by persons with dyslexia or other reading and writing difficulties. The complete program was also to contain a spell checker, which was developed especially for persons with dyslexia. The two active components, the predictor and the spell checker, shared some resources and were able to interact with one another. The following sections describe the motivation for the probabilistic design of the word predictor and the language model employed.

Design of the New Predictor

The major design goal for the new predictor was that it should suggest noticeably more appropriate words than the old predictor and, in particular, that it should suggest words that are grammatically probable. A realistic goal is a word predictor that suggests more appropriate words before less likely words. This goal can be achieved by the use of a probabilistic algorithm.

A probabilistic treatment of language modeling frequently used in speech recognition systems was developed at IBM (Jelinek, 1990). This model uses word and word class n-grams (some integer, n, of words or word classes in a sequence). Jelinek (1991) remarked that, after 15 years progress in speech recognition, the (word) trigram model has remained fundamental. Reasons given for its success include the following: (a) trigrams are firmly based on data, so the more, the better; (b) trigrams simultaneously reflect syntax, semantics, and pragmatics of the domain; and (c) European languages have a strong tendency toward standard word order and are thus substantially local. However, trigram models do have problems (Jelinek, 1991). They are not as suitable for highly inflected languages because a much larger vocabulary is required and, therefore, a much larger corpus is necessary. Jelinek conjectured that a substantially better alternative to word trigrams would be based on a grammar-related approach, with substantial components of the grammar itself derived automatically from text corpora. To derive this grammatical information, text corpora must be labelled with the word class of each word in the text.

A probabilistic language model including word bigrams and word class trigrams has been employed for the new version of Prophet (see below). A large training text has been used to extract language statistics for both words and word classes.

The Language Model

Two Markov models are employed in the language model, one for words and one for word classes. A Markov model is an effective way of describing a stochastic (probabilistic) chain of events, such as a string of words. Such a model consists of a set of states and probabilities of transitions between them. The last *m* words in a string are effectively remembered by an *m*th-order Markov model. Using the previous word in a text required a first-order Markov model; a second-order Markov model was used to take advantage of knowledge of the previous two word classes.

The order of the word model was determined by the size of the required database. More information could be brought to bear if the two previous words were considered, as recommended by Jelinek (1991), but this would require word trigrams to be extracted from the training text and stored in a database. However, such a database could take an unreasonable amount of storage space and calculation time for the program, which was written for use with other programs on a personal computer. As the number of n-grams increases, there are fewer repetitions of any particular n-gram (i.e., more unique n-grams); thus, more storage space is needed and more calculation time is required when the algorithm is in operation.

Fewer repeated n-grams also lead to what is known as "sparse data problems," in particular, lack of representativeness for other texts.

The order of the word class model was determined by experiment (Carlberger, Magnuson, Carlberger, Wachtmeister, & Hunnicutt, 1997) and by the fact that the number of word class trigrams is quite manageable. The experiment, testing six texts (ranging in size from 1,000 to 10,000 words) within the four conditions of no word class labels, unigrams, bigrams, and trigrams found bigram word class labels to produce substantial savings in keystrokes and trigrams to provide a marginal increase over bigrams.

As an example in Swedish of the phenomena word trigrams cover, consider all sequences such as *en lätt uppgift, en svår uppgift*, and *någon svår uppgift* (an easy task, a difficult task, and some sort of difficult task). After the two first words have been typed, the predictor should suggest *uppgift* (task) before *uppgifter* (tasks). To do so, it suffices for the predictor to have the information that the two previous words were a singular determiner and an adjective. The class trigram is simply (1) singular determiner, (2) adjective, and (3) singular indeterminate noun. It is only one entry in the word class database instead of thousands of entries of possible word combinations in a word database.

The word and word class models interact, but the separation enables the predictor to work with lexicons of either grammatically classified or grammatically unclassified words without any changes to the program. (This can facilitate future localizing into languages for which no grammatically labelled texts may be available.) One first obtains a probability estimation for the word class of the next word, using the word class Markov model, and obtains a probability estimation for the next word given the probability estimate of its word class and given the previous word, using the word Markov model. Thus, the word class probability estimation is taken into account to promote words that have a likely word class.

Using Word Class Tags

To construct such a database of word class bigrams and trigrams, a training text is needed in which the words are tagged, which means that they are classified according to their word class. The process of assigning tags to the words of a text is called tagging, and a computer program that tags a text is consequently called a tagger. An example of a tagged text with simple grammatical classes could be "*This* <article> *is* <verb> *a* <article> *tagged* <adjective> *sentence* <noun>."

For the new predictor, a tagged corpus (text collection) was used to train a tagger developed at the Universities of Stockholm and Umeå (Ejerhed, Källgren, Wennstedt, & Åstrom, 1992). The tag set consists of about 150 word class tags. The level of detail of these tags is illustrated by nouns, which are subdivided according to gender, plurality, definiteness, and (sometimes) case. For example, the word *åker* (a field where crops are grown) is tagged <*noun neuter singular indefinite*>. The tagged corpus consists of 1 million words, yielding an average tag trigram count of 12.8 (i.e., the average number of times a particular word class trigram occurs in the corpus is 12.8). In cases in which a particular trigram has occurred only rarely or not at all, tag bigrams and unigrams are weighted more (Jelinek, 1990). In cases in which the tagging is ambiguous (i.e., tagged words had more than one possible word class), an average is calculated.

Managing Unknown Words

The words known to the predictor when a training session starts have statistical information associated with them. There are counts for word unigrams and word bigrams and counts for how many times each word has been tagged with different tags in the training text. No such information is available for the new words encountered during a prediction session. It is an advantage if this information for new words can be extracted as they are used, just as information was once retrieved from the training text. However, one does not want to change the database constructed from the training text since it would be very troublesome to keep it statistically consistent. New words are therefore stored separately from the previously known words of the database. A method was investigated for deducing a likely tag by statistically analyzing the morphology of words with respect to their tags. Since the last few letters of a word reveal much about its word class, a limited set of word endings was sought that would give as much information as possible about the probability distribution of tags associated with words having common endings. An algorithm to find such a set of suffixes was devised.

EVALUATION

The quality of the language model has been evaluated by using it to predict the words of another, smaller test text. The results of these simulations are given in terms of the ability of the algorithm to reduce the number of keystrokes needed for a user to compose a text. The contribution of the tag Markov model is examined as well as keystroke savings gained by prediction of the word class of unknown words. A number of heuristic improvements are then investigated to determine the degree to which the number of keystrokes can be reduced. These include recency promotion, repetition of predictions, case sensitivity, and the derivation of inflected forms of words for a user to choose among.

Materials

The corpus used for the study was the Stockholm Umeå Corpus (SUC), which contains about 1 million running words (Ejerhed et al., 1992). This corpus is tagged with about 150 word classes.

Procedure

Using an n-gram extractor developed for the purpose, a set of training texts containing about 10,000 words from the tagged SUC was scanned, and occurrences of word n-grams and tag n-grams were stored in files. A generation program developed for the purpose of constructing a database was then used to prune the sets of n-grams and to sort them. Additional information, including inflectional category, was added where appropriate, and the information was stored on file as a main lexicon. This main lexicon was subsequently accessed by the prediction algorithm.

In a simulation program, optimum weights of the Markov model formulas and the parameters of the heuristic modifications were calculated. This simulation program was given three new texts and used the algorithm to reproduce the text with as few keystrokes as possible. The simulator represents a perfect user who does not make any typing errors and who does not miss any correct predictions. When the text was completed, the simulator responded with the following data:

- Text statistics, such as the number of letters and words in the text;
- Text coverage (i.e., the fraction of running words that were known to the predictor before the simulation started);
- Letter and overall keystroke savings; and
- The distribution of the positions of the correct predictions in the prediction list.

This information was then used to evaluate the performance of different lexicons and configurations of the algorithm.

The definition of keystroke savings used is a common one for direct selection: the percentage of keystrokes eliminated (i.e., saved) in the test texts by employing word prediction with standard keyboard input instead of standard keyboard input alone. This measure was chosen not because it gives a total picture of the help provided a user but rather because it is a simple measurement that gives a commonly employed and well-understood measure of the success of the algorithm itself, and of its associated lexicons, in making appropriate predictions.

The word Markov model, with unigrams and bigrams, was implemented in all trials as a base algorithm. Calculations were made based on one character per typed letter and one character per typed space, punctuation mark, shift, or function key. Inflectional variations, which are usually available, were removed from the field of choices. Tests were carried out for both one and five predictions. The language was Swedish.

Computer Language and Equipment

All programs are written in ANSI C++ to ensure portability. A number of generic classes, such as hash tables, lists, arrays, and object pools, have been developed to handle sets of objects with varying demands on access to their elements. These classes have been reused extensively in all parts of the programs. The programs are modular to enable different types of functionality to be well isolated from each other. This facilitates easy implementation of future improvements.

Program development was carried out at a work station that ran UNIX. Tests were also run in this environment. The code has also been successfully exported to be integrated with a user interface on a personal computer that was being developed by another partner in the project.

Results of Simulations

Four independent 10,000-word texts were used for the experiments—one text for optimizing the parameters and the other three for simulation and evaluation. The tag (word class) Markov model, adaptivity by using topic lexicons, and heuristic modifications of the language model all contribute to improve the predictions. By comparing the keystroke savings of an optimized configuration with the keystroke savings of the same configuration, but with one feature removed, the impact of that feature is revealed. Table 1 lists each of the features, the percentage of keystroke savings attained without that feature, and the increase in keystroke savings obtained when that feature is added to an otherwise optimal configuration. Note that the sum of these complementary values is always equal to the value for keystroke savings given in the last row when all features are employed in the algorithm.

Tag Model

The implementation of the tag Markov model increased the keystroke savings by 4.2% for one prediction and by 2.8% for five predictions. The higher figure for a single prediction shows the power of the tag model in providing a better first choice.

Word Learning

The implementation of word learning and word class hypothesizing for new words by the program increased the keystroke savings by 3.2% for one prediction and by 5.1% for five predictions.

Recency Promotion

To implement recency promotion, an experimentally derived additional term was included in the probability function, which consists of three factors. One factor has two different values, depending on whether the word is a function word or a content word; this reflects the fact that the reappearance of content words is meaningful in the text. The second factor is

TABLE 1: **Percentage of Keystroke Savings by Feature for One and Five Predictions**

	One Prediction		Five Predictions	
Feature	*KS Saved, No Feature*	*Increase, with Feature*	*KS Saved, No Feature*	*Increase, with Feature*
Tagging	28.7	4.2	43.2	2.8
Learning new words	29.7	3.2	40.9	5.1
Recency promotion	31.9	1.0	43.9	2.1
Repetition delay	29.2	3.7	45.4	0.6
Case sensitivity	32.4	0.5	45.8	0.2
Autoinflection	32.8	0.1	45.7	0.3
All features	32.9		46.0	

KS = keystrokes.

a value ranging from 0 to 255 that is initially 0 and that is increased by a small amount each time a particular word occurs. The third factor is a probability function of the tag of the word under consideration, depending on the tag trigram it completes, thus making recency promotion sensitive to the previous words so that words with appropriate tags are promoted to a greater extent than other words. The implementation of this heuristic modification increased the keystroke savings by 1.0% for one prediction and by 2.1% for five predictions.

Repetition of Predictions

When the user types letters that make the prefix (i.e., the part of a word already typed) match a word already suggested, it is possible either to suggest the same word repeatedly or to disregard it. For example, while typing the first letters of the word jagar (hunts), the predictor may well suggest *jag* (the pronoun "I") three times, thereby delaying the prediction of *jagar*. This behavior is preferable when the user misses selecting a correct prediction because he or she will soon have another opportunity to select the same word. On the other hand, multiple occurrences of the same word decrease the number of other words that can be suggested, hence decreasing the possible keystroke savings.

Some users may benefit from having predictions repeated early, whereas others may not. Therefore, it was decided to implement a user option that delays the repetition of predictions until all other matching words have been suggested. This is accomplished by scaling down the probabilities of words that have already been suggested during the typing of a particular word. This facility brings about a substantial improvement in keystroke savings for one prediction (3.7%), where reprediction precludes prediction of other words, but only a small improvement for five predictions (0.6%).

Case Sensitivity

In addition to automatic capitalization (of the initial word in a sentence and of proper nouns in a prediction list), there is another way to improve the quality of predictions by monitoring the user's employment of uppercase letters in a word. If the user capitalizes the initial letter of a word, the predictor can promote words that are usually spelled with an initial capital, typically proper nouns. Conversely, if the user does not capitalize the initial letter, the predictor can promote words whose initial letter is normally written in lower case. This promotion is achieved simply by scaling down the probabilities for the capitalized or noncapitalized words. With this enhancement, a very slight increase in keystroke savings was recorded. Keystroke savings for one prediction increased by 0.5% and for five predictions by 0.2%. The perceived improvement in prediction quality is greater than these small improvements show, however, due to clearly more appropriate predictions when a capital letter is typed to begin a new word.

Deriving Inflected Forms of Words

The main lexicon in the new system contains both base forms and inflected forms of words; for example, it contains the base form of the strong verb *göra* (to do) and the inflected forms *gjorde* (did) and *gjort* (done). Each word in the lexicon that is the base form of an adjective, verb, or noun is marked with an inflection rule category. Accessing the rule category and possibly the last letters in the base form permits all inflected forms to be derived. The benefit of using this approach is the accessibility of many inflected forms with a frequency of occurrence lower than the minimum needed for inclusion in the main lexicon, both for the purpose of keystroke savings and when grammatical aid is required.

The inflected forms of words are derived in two different situations. Just as in earlier versions, words in

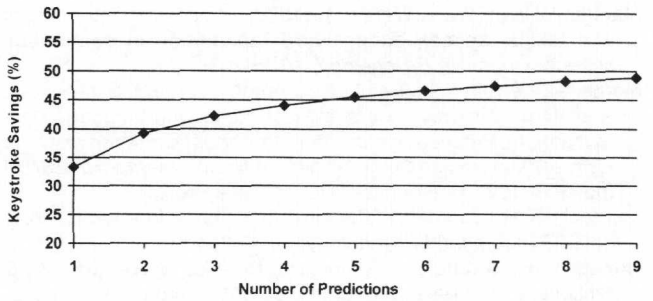

Figure 1. Keystroke savings as a function of prediction list length.

the prediction list that can be inflected are marked, which enables the user to obtain another list with the inflected forms, when desired. Simulation of this feature is quite complex and was not attempted. In the new version, inflected words can also be automatically derived when the predictor runs out of matching words. This feature was found to increase keystroke savings very slightly in simulations. The keystroke savings for one prediction was 0.1% and for five predictions 0.3%. Once again, the perceived improvement is greater than these figures show. In case the user is unsure of the spelling of an inflected form, it can be easily accessed, always appearing in the same order in the inflection list. When further predictions are otherwise unavailable, the presentation of inflected forms is an advantage.

Number of Predictions

The number of words in a prediction list has a large impact on keystroke savings, which appears to increase monotonically with the number of alternatives. Venkatagiri (1994) found a 7% difference in keystroke savings between 5 and 10 predictions and a 6% difference between 10 and 15 predictions. Swiffin et al. (1987b) found a difference of about 3% between both 5 and 10 predictions and 10 and 15 predictions with an initially empty lexicon and a difference of about 4% between 5 and 10 predictions with a prebuilt dictionary. However, the more predictions there are, the longer the time it takes for the user to inspect them and the greater the chance of missing correct ones. This observation has been made by a number of researchers (see, for example, Soede & Foulds, 1986). Also, as the number of predictions grows, the increase in keystroke savings diminishes (i.e., the keystroke saving plateaus). For English, there is a "knee" at about 5 predictions (Swiffin et al., 1987b). It has been our experience that most users are able to quickly scan 5 alternatives, which is also the default setting in Prophet. The percentage of keystroke savings for 1 to 9 predictions is shown in Figure 1. For 9 predictions, the keystroke savings are approximately 50%. The difference in keystroke savings, comparing the number of predictions differing by 5 (e.g., comparing 3 with 8 and 4 with 9), is similar to that found by Venkatagiri (1994), between 6% and 8%.

CONCLUSIONS

The results of the current project revealed that the new version of the Swedish word prediction program with new unigram and bigram word lexicons of about 10,000 entries each provides approximately 46% in keystroke savings with five predictions. When using nine predictions, the new version achieves keystroke savings of about 50%. It is to be expected that larger lexicons than those that were used in this study would also increase keystroke savings by several additional percentage points.

The current project also demonstrated that the use of a tag Markov model with class trigrams contributed most to keystroke savings (a 4.2% increase) in the case of single predictions. Single predictions are rarely used in today's word prediction programs, however, and are perhaps employed only when the goal is to achieve a very low cognitive load for the user. The corresponding increase for five predictions was only 2.8%. The facility for learning new words provided by the program contributed most for five predictions, a 5.1% increase from 40.9% to 46.0%. The promotion of recently used words made the third substantial contribution with five predictions, increasing keystroke savings by 2.1%. For one prediction, however, recency promotion accounted for only 1% of keystroke savings, clearly being eclipsed by repetition delay (i.e., delaying the repetition of an unchosen predicted word), with a contribution of 3.7%. Limiting prediction to words beginning with an upper- or a lowercase letter (depending on which case is typed word-initially) and automatically predicting inflected forms of words when there are no more words from the lexicon to predict added less than 1% each in keystroke savings for both one and five predictions. The question of whether the syntactic knowledge and heuristics employed in this algorithm and in other word prediction systems can, as assumed, prove to be noticeably beneficial to users despite the rather low keystroke savings remains to be investigated.

ACKNOWLEDGMENTS

We would like to express our appreciation to the AMS (National Labour Market Board), the RFV (National Social Insurance Board), and The Swedish Handicap Institute for supporting this research.

Address reprint requests to: Sheri Hunnicutt, Department of Speech, Music, & Hearing, Kungliga Tekniska Högskolan, Dr. Kristinasvag 31, S-100 44 Stockholm, Sweden.

REFERENCES

Boekestein, M. (1996). *Word prediction.* Unpublished master's thesis, Catholic University of Nijmegen, Nijmegen, Sweden.

Bradley, D. (1978). *Computational distinctions of vocabulary type.* Unpublished doctoral dissertation, Massachusetts Institute of Technology, Cambridge, MA.

Broadbent, D. E. (1967). Word-frequency effect and response bias. *Psychological Review, 74,* 1–15.

Carlberger, A., Magnuson, T., Carlberger, J., Wachtmeister, H., & Hunnicutt, S. (1997). *Probability-based word prediction for writing support in dyslexia.* (PHONUM, pp. 17–20). Stockholm, Sweden: Department of Phonetics, Umeå University.

Carlson, R., Elenius, K. J., Granström, B., & Hunnicutt, S. (1985, January). Phonetic and orthographic properties of the basic vocabulory of five European languages. *Speech Transmission Laboratory/Quarterly Progress and Status Report.*

Ejerhed, E., Källgren, G., Wennstedt, O., & Åstrom, M. (1992). *The linguistic annotation system of the Stockholm-Umeå corpus project* (DGL-UUM-R-33, Report No. 33). Stockholm, Sweden: Department of General Linguistics, Umeå University.

Gibler, C. D., & Childress, D.S. (1983). Adaptive dictionary for computer-based communication aids. In *Proceedings of the Sixth Annual Conference of the Rehabilitation Engineering Society of North America (RESNA)* (pp. 65–167). Washington, DC: RESNA Press.

Heckathorne, C. W., Leibowitz, L., & Strysik, J. (1983). MicroDEC II: Anticipatory computer input aid. In *Proceedings of the Sixth Annual Conference of the Rehabilitation Engineering Society of North America (RESNA)* (pp. 43–36). Washington, DC: RESNA Press.

Heckathorne, C. W., Voda, J. A., & Leibowitz, L. J. (1987). Design rationale and evaluation of the portable anticipatory communication aid: PACA. *Augmentative and Alternative Communication, 3,* 170–180.

Higginbotham, D. J. (1992). Evaluation of keystroke savings across five assistive communication technologies. *Augmentative and Alternative Communication, 8,* 258–272.

Hunnicutt, S. (1986). Lexical prediction for a text-to-speech system in communication and handicap. In E. Hjelmquist & L. G. Nilsson (Eds.), *Aspects of psychological compensation and technical aids* (pp. 253–263). Amsterdam: Elsevier Science.

Hunnicutt, S. (1987). *Input and output alternatives in word prediction.* (STL-QPSR 2-3/1987). Stockholm, Sweden: Department of Speech, Music and Hearing, Kungliga Tekniska Högskolan.

Hunnicutt, S. (1989). Using syntactic and semantic information in a word prediction aid. In *Proceedings of Eurospeech: Vol. 1* (pp. 191–193). Paris: CEP Consultants.

Hunnicutt, S., Bertenstam, J., & Raghavendra, P. (1994). A morphologically-based word predictor for Swedish. In *Proceedings of the 17th Annual Conference of the Rehabilitation Engineering Society of North America (RESNA)* (pp. 106–108). Washington, DC: RESNA Press.

Jelinek, F. (1990). Self-organized language modelling for speech recognition. In A. Waibel & K. F. Lee (Eds.), *Readings in speech recognition* (pp. 450–506). San Mateo, CA: Morgan Kaufmann.

Jelinek, F. (1991). Up from trigrams! The struggle for improved language models. In *Proceedings of Eurospeech: Vol. 3* (pp. 1037–1040). Geneva: European Speech Communication Association.

Kucera, H., & Francis, W. N. (1967). *Computational analysis of present day English.* Providence, RI: Brown University Press.

Lesher, G. W., Moulton, B. J., & Higginbotham, D. J. (1998). Techniques for augmenting scanning communication. *Augmentative and Alternative Communication, 14,* 81–101.

Light, J., Lindsay, P., Siegel, L., & Parnes, P. (1990). The effects of message encoding techniques on recall by literate adults using AAC systems. *Augmentative and Alternative Communication, 6,* 184–201.

Marslen-Wilson, W., & Welsh, A. (1978). Processing interactions and lexical access during word recognition in continuous speech. *Cognitive Psychology, 10,* 29–63.

Morris, C., Newell, A. F., Booth, L., & Arnott, J. (1991). Syntax PAL: A system to improve the syntax of those with language dysfunction. In *Proceedings of the 14th Annual Conference of the Rehabilitation Engineering Society of North American (RESNA)* (pp. 105–106). Washington, DC: RESNA Press.

Murdock, B. B., Jr. (1962). The serial position of free recall. *Journal of Experimental Psychology, 64,* 482–488.

Newell, A. F., Arnott, J. L., Booth, L., Beattie, W., Brophy, B., & Ricketts, I. W. (1992). Effect of the "PAL" word prediction system on the quality and quantity of text generation. *Augmentative and Alternative Communication, 8,* 304–311.

Palazuelos-Cagigas, S., Godino-Llorente, J., & Aguilera Navarro, S. (1997). Comparison between adaptive and non-adaptive word prediction methods in a word processor for motorically handicapped non-vocal users. In G. Anogianakis, C. Bühler, & M. Soede (Eds.), *Advancement of assistive technology* (pp. 120–124). Amsterdam: IOS Press.

Palazuelos, S., Aguilera, S., Ricketts, I., Gregor, P., & Claypool, T. (1998a). Artificial neural networks applied to improving linguistic word prediction. In *Proceedings of the Biennial Conference of the International Society for Augmentative and Alternative Communication* (pp. 193–194). Dublin: ISAAC.

Palazuelos, S., Aguilera, S., Rodrigo, J., & Godino, J. (1998b). Grammatical and statistical word prediction system for Spanish integration in an aid for people with disabilities. In R. H. Mannell & J. Robert-Ribes (Eds.), *Proceedings of the 5th International Conference on Spoken Language Processing* (pp. 2479–2482). Sydney: Australian Speech Science and Technology Association.

Soede, M., & Foulds, R. A. (1986). Dilemma of prediction in communication aids and mental load. In *Proceedings of the Ninth Annual Conference on Rehabilitation Technology* (pp. 357–359). Washington, DC: RESNA Press.

Swiffin, A. L., Arnott, J. L., & Newell, A. F. (1987a). The use of syntax in a predictive communication aid for the physically handicapped. In *Proceedings of the Tenth Annual Conference of the Rehabilitation Engineering Society of North America* (pp. 124–126). Washington, DC: RESNA Press.

Swiffin, A., Arnott, J., Pickering, A., & Newell, A. (1987b). Adaptive and predictive techniques in a communication prosthesis. *Augmentative and Alternative Communication, 3,* 181–191.

Swiffin, A. L., Pickering, J. A., Arnott, J. L., & Newell, A. F. (1985). PAL: An effort efficient portable communication aid and keyboard emulator. In *Proceedings of the Eighth Annual Conference on Rehabilitation Technology,* (pp. 197–199). Washington, DC: RESNA Press.

Tyvand, S., & Demasco, P. (1993). Syntax statistics in word prediction. In *Proceedings of the European Conference on the Advancement of Rehabilitation Technology (ECART2)* (p. 11.2). Stockholm: The Swedish Handicap Institute.

Vandyke, J., McCoy, K., & Demasco, P. (1992). Using syntactic knowledge for word prediction. In *Proceedings of the Fifth Biennial Conference of the International Society for Augmentative and Alternative Communication* (p. 175). Philadelphia: ISAAC.

Venkatagiri, H. (1994). Effect of window size on rate of communication in a lexical prediction AAC system. *Augmentative and Alternative Communication, 10,* 105–112.