The Keystroke Savings Limit in Word Prediction for AAC

Keith Trnka, Debra Yarrington, Kathleen McCoy
Computer Science Department
University of Delaware
Newark, DE 19716
{trnka,yarringt,mccoy}@cis.udel.edu

ABSTRACT

Word prediction is a method for enhancing the communication ability of persons with speech and language impairments. In this work, we demonstrate that two measurement settings affect evaluation of word prediction systems significantly, and that consideration of these settings is crucial to accurate interpretation of results.

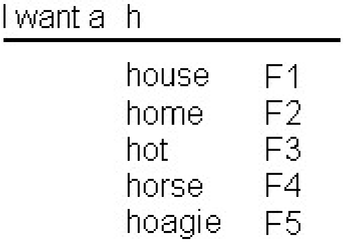
Keywords
Word prediction, keystroke savings, alternative and augmentative communication (AAC)

INTRODUCTION

Alternative and Augmentative Communication (AAC) is the field of research concerned with finding ways to help those with speech difficulties communicate more easily and completely. Today there are approximately 2 million people in the United States with some form of communication difficulty. With today's technology, electronic communication devices are widespread. Most communication devices today have speech output and/or displayed text or pictures. One issue in using a communication device is that communication rate is generally slower than the common speaking rate. Whereas speaking rate is estimated at 180 words per minute (wpm) and experienced typists can manage 100 wpm, a disabled user's input rate is estimated at roughly 15 wpm [2,5,21]. Thus one goal of developers of AAC devices is to find ways to increase the rate of communication output. Developers cannot increase the dexterity or muscle control of users, so the alternative is to experiment with the user interface for input.

This paper investigates the use of a word prediction system. In word prediction, we assume that the user enters letters using a standard keyboard. The system predicts full words that are likely to be desired, and provides them to the user for selection with one additional keystroke.

A word prediction system predicts the word currently being typed on the basis of what has already been typed. Suppose that the user wants to enter "I want a home in the country." After typing, "I want a h", they might see something like shown below. The system has created a *prediction window* containing the five words which it thinks the current word is most likely to be. In this example, the user can press F2 to complete the word "home" and the system will automatically enter a space afterwards. So in this example, the user needed 3 keystrokes to enter what would normally take 5 keystrokes, when the space is considered.



```
I want a  h

          house    F1
          home     F2
          hot      F3
          horse    F4
          hoagie   F5
```

The prediction list can vary in length, but most systems tend to use lists of length between five and seven. The prediction list can occur in-line (it can appear within the line being entered as it is entered), or it can occur somewhere separately on the interface screen. For row-column scanning devices, the word list often appears in an extra row or column on the scanning grid.

It is difficult to judge how much word prediction can speed communication rate. Much of this determination is dependent on both the characteristics of the user, such as their physical and cognitive abilities, and characteristics of the user interface, such as where the prediction list is displayed and how a word in the list is selected.

It quickly becomes apparent that many factors affect the efficacy of word prediction, and more studies are needed to determine the effect different factors have on the success of word prediction. It is equally apparent, however, that unless the word prediction system is able to successfully predict words and thus decrease the number of keystrokes necessary, other factors are irrelevant. Therefore, before tackling the added issues involved in user interfaces, it is instructive to look at the percentage of keystrokes saved, since this measure provides an upper bound on any communication rate increases from word prediction. Thus our work here concentrates on investigating keystroke savings in word prediction.

Our long-term goal is to investigate methods for increasing keystroke savings in word prediction by taking various amounts of contextual information into account during the prediction process. Of course, in doing this we must have a way of evaluating whether or not our various attempts at capturing contextual information are fruitful - so first we must establish a baseline prediction system and a method for calculating keystroke savings against which our future systems can be tested.

Clearly this paper is not the first to discuss the use of word prediction in AAC [3,4,5,6,8,13], and each of these systems has been presented with evaluations. On the other hand, upon closer inspection of the literature we find that it is difficult to judge the performance gains of such systems against each other because the previous work has often been unclear about exactly what the assumptions are underlying their keystroke savings calculation and because the theoretical limit of keystroke savings has not been established (but see [5,15] for attempts at this).

In this paper we first give some background in statistical approaches to word prediction. We next discuss some issues in evaluating keystroke savings that must be addressed. We point out how making different assumptions about the user interface (e.g., whether the "speak" key is included in keystroke calculations) can change the keystroke savings of a system dramatically. We present an evaluation of our baseline system (making all such assumptions explicit). **Finally we present some future directions with insights into how we anticipate incorporating context into our baseline system, and conclude.**

METHODS

Like other approaches, we apply statistical language modeling techniques to word prediction [3,4,5,6,8,13]. The approach presented here a pure n-gram based method. [5,6,8] additionally integrated syntactic knowledge into their language models, which was found to improve prediction somewhat. [13] added topic modeling onto an n-gram approach, which also had mixed results.

Basic word prediction treats each sentence of the user's conversation as independent. At any given point in the sentence, the user has some word that they are typing. At that point, the word prediction system presents a list of words, called a *prediction window*, that the user may be typing. If the desired word appears in the list, the user selects it using a key reserved for that position in the list. If the word doesn't appear in the list, the user must enter another character. Then the word prediction system updates the list and the process repeats.

To present the user with a list of possible words, the word prediction system needs to know all of the words in the language. The vocabulary is constructed by considering all words that occur in some training corpus. If a word being typed isn't in the vocabulary, it can't be predicted.

The second requirement is a statistical language model. The purpose of the language model is to compute the probability P(word | history), where the history is the words that have already been entered. Given a vocabulary and a language model, the list of predictions is generated as follows: The vocabulary is first filtered to remove words that don't match the partially entered word. For example, if 'a' has been entered, the system will only consider words beginning with 'a'. Then this list of candidates is sorted by P(word | history). The top W words are presented to the user, where W is the prediction window size.

Specifically, in this work, we train a trigram model with backoff on part of the Switchboard corpus and test on the remainder of Switchboard. The backoff weights are computed using Good-Turing smoothing and a special unigram model is used for the first word of a sentence. For more information on this language model, refer to **[[LM tech report]].**

EVALUATION

Evaluation of word prediction is performed using a simulated user interface and a simulated user. The user interface simulates prediction using window sizes of 1-10, and adds a space after the word when prediction is used. Unless stated otherwise, one character must be entered before prediction is attempted, called *delayed prediction*.

The simulated user is assumed to be perfect in the sense that the user selects the desired word as soon as it appears in the prediction window and that the user decides not to select a word from the window if normal typing would be faster. For example, if the user typed “an” for the word “an”, using word prediction would yield no benefit, because the user would either enter a space without word prediction, or select the prediction. All evaluation is done using keystroke savings (KS):

$$KS = \frac{keys_{normal} - keys_{with\ prediction}}{keys_{normal}} \times 100$$

Finally, the simulated user has decided what to say ahead of time: all of the text from the testing portion of the Switchboard corpus.

End of Sentence

One of the seemingly minor details of the simulated user interface that we've found is in ending a sentence. To signal that the sentence or utterance is complete, a user would press a “speak key” (SK). However, we are not aware of any researchers that simulated this. The difference is that the speak keystrokes cannot be reduced using prediction, so measuring keystroke savings without simulating a keystroke to speak is an overestimate of the benefit of word prediction.

Window Size	Without SK	With SK
1	37.1	36.3
5	50.6	49.7
6	51.6	50.6
10	54.1	53.2

At a window size of 1, the reduction in savings is 0.8%. The reduction increases slightly with window size to peak at 1%. The difference in keystroke savings between these two simulations might not seem important, but it is substantial when compared to other changes in keystroke savings: [6] improved keystroke savings by 0.9% by considering part-of-speech. At their window size of 5, their improvement is equal to the error one would have in not counting the speak key. [14] found that increasing the training text size from 1 million words to 3 million words on a trigram approach yielded an increase of 2.5% savings. They also found that the increased savings of using a trigram model over a bigram model was 0.8% savings. In comparison to the size of these improvements, this difference in measurement is substantial. Consideration of a keystroke to end a sentence makes the simulation more realistic, so all further evaluations in this paper include the speak keystroke.

The Limit of Keystroke Savings

Copestake [5] and Lesher at. al. [15] have both discussed the limits of word prediction. Copestake noticed that it was difficult to practically exceed 50% keystroke savings using text produced by an AAC user. She subsequently applied Shannon's estimate of the entropy of English to estimate that 50-60% keystroke savings is a realistic limit of word prediction. [15] also began with the observation that increases in keystroke savings become exceedingly difficult beyond a certain point. For their work, 55% keystroke savings was this practical barrier. They subsequently had humans provide the prediction windows on some Switchboard texts, and found that the group of humans achieved 59% keystroke savings on average. However, one human neared 70% savings on some individual texts. Lesher et. al. conclude that better word prediction is still possible.

While reasoning about the keystroke savings metric, we noticed that it isn't possible to achieve 100% keystroke savings. That would require the system to enter the entire conversation without a single key!

We've measured the theoretical limit of word prediction on our test set by simulating a perfect word prediction engine – one that predicts the correct word as early as possible. When prediction requires that one character be entered, the maximum possible keystroke savings for our test corpus is 58.4%. This corresponds to two keystrokes per word: one to type the first letter and one to select the prediction. Using these user interface settings, actual word prediction barely exceeds 55%, so there may be a practical limit that is less than the theoretical one. See below for a comparison between the limit and actual savings.

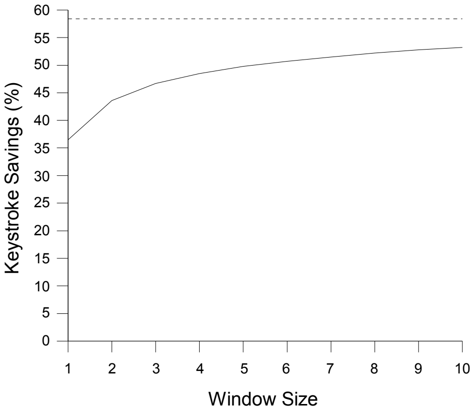


Figure 2: Keystroke savings approaches the theoretical limit as window size is increased

Is the practical limit of word prediction related to the theoretical one? To investigate this, we measured the theoretical limit and actual performance of word prediction when the prediction window is available before any characters are typed, called *immediate prediction*. This cuts the minimum number of keystrokes required in half and raises the theoretical limit of keystroke savings to 80%! See Figure 3 for the effect on our word prediction method. The limits of both approaches are shown for reference.

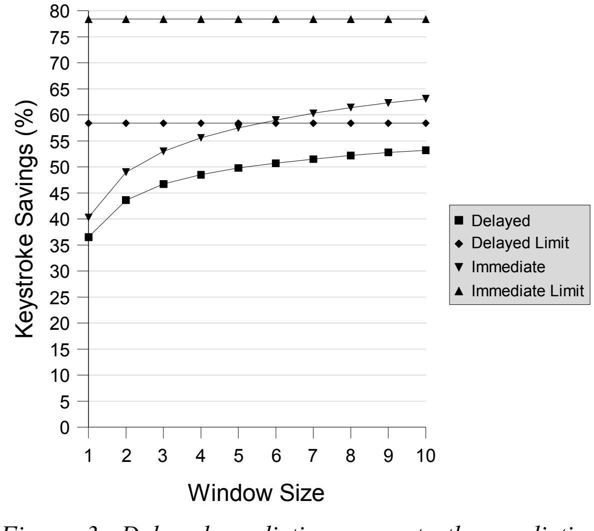


Figure 3: Delayed prediction presents the prediction list to the user after one character is typed; immediate prediction presents the list before any have been typed.

The results show several interesting trends. Most notably, waiting a character before presenting a prediction list to the user sacrifices 4-10% keystroke savings, depending on window size. By comparison, increasing the window size of delayed prediction from 2 to 6 increases keystroke savings by 7%. The difference between delayed and immediate prediction increases with window size because the limit of delayed prediction becomes more significant: at a window size of 1, delayed prediction often needs more than one character to correctly guess the word being typed. However, at window size 10, the desired word is often in the prediction list after the first letter has been entered. So at window size 10, many words are entered in two keystrokes. Of these words, many could have been entered using a single keystroke if immediate prediction had been used.

The second notable aspect of the graph above is that immediate prediction is nowhere near it's theoretical limit. If we look at each approach in terms of realized potential (RP), where

$$RP = \frac{keys_{\text{normal}} - keys_{\text{with prediction}}}{keys_{\text{normal}} - keys_{\text{minimum}}} \times 100$$

then we find that delayed prediction at window size 5 has a realized potential of 87% and immediate prediction at the same window size has a realized potential of 73%. In other words, prediction without any letters of the word is more difficult, so there is more room for improvement.

Finally, the keystroke savings achieved by our word prediction method are significant. However, for discussion of this, refer to **[[LM tech report]]**.

FUTURE WORK

Although we have investigated two of the user interface settings that affect keystroke savings, there may be more such settings we have not investigated. We have discovered a third setting that is primarily relevant for the application of a word prediction system for written text, rather than spoken text. The punctuation of written text can rarely be predicted, so the keystroke savings of a system that ignored punctuation in evaluation would be inflated. Finally, punctuation has a profound impact on whether it is beneficial for the system to include a space after every word automatically.

CONCLUSIONS

In crafting a baseline for future work, we found that results between different word prediction systems were not comparable due to differences in measurement and testing data. By exploring the theoretical limitations of word prediction, we've demonstrated that details of the word prediction environment are crucial to evaluation: a 1% improvement is significant when only 5% improvement is possible. The same 1% improvement would be insignificant when 30% improvement is possible. Some researchers have used immediate prediction [3,15] and others have used delayed prediction [5,4]. However, [6,8,13] did not specify, which makes their results difficult to interpret. Similarly, no research cited in this work has described whether or not a speak key is included. In conclusion, an appreciation of the impact of user interface decisions on evaluation is necessary both to comparing results between researchers and improving the field of word prediction.

ACKNOWLEDGMENTS

We would like to thank the US Department of Education for funding this research under grant H113G040051, under the National Institute on Disability and Rehabilitation Research program. Also, thanks to Gregory Lesher for correspondence regarding his work.

REFERENCES

1. Bellegarda, Jerome. Large Vocabulary Speech Recognition with Multispan Language Models. *IEEE Trans. On Speech and Audio Processing*, 8(1): 2000.
2. Beukelman, D. R. and Mirenda, P. *Augmentative and alternative communication: Management of severe communication disorders in children and adults*. Baltimore: Paul H. Brookes Publishing Co., 1998.
3. Boggess, Lois. Two simple prediction algorithms to facilitate text production. *Proceedings of Applied Natural Language Processing*, 1988.
4. Carlberger, Alice, John Carlberger, Tina Magnuson, M. Sharon Hunnicutt, Sira Palazuelos-Cagigas, and Santiago Aguilera Navarro. Profet, A New Generation of Word Prediction: An Evaluation Study. *Proceedings of Natural Language Processing for Communication Aids*, 1997.
5. Copestake, Ann. Augmented and alternative NLP techinques for augmentative and alterative communication. *Proceedings of Natural Language Processing for Communication Aids*, 1997.
6. Fazly, Afsaneh and Graeme Hirst. Testing the Efficacy of Part-of-Speech Information in Word Completion. *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, 2003.
7. Florian, Radu and David Yarowsky. Dynamic Nonlocal Language Modeling via Hierarchical Topic-Based Adaptation. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, 1999.
8. Garay-Vitoria, Nestor and Julio González-Abascal. Intelligent Word-Prediction to Enhance Text Input Rate. *Proceedings of the second international conference on Intelligent User Interfaces*, 1997.
9. Hindle, Donald. Deterministic Parsing of Syntactic Non-fluencies. *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, 1983.

10. Jurafsky, Daniel and James Martin. *Speech and Language Processing*. Prentice Hall, Upper Saddle River NJ, 2000.
11. Katz, Slava. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Trans. On Acoustics, Speech, and Signal Processing*, 35(3): 1981.
12. Koester, H.H. and Levine, S.P. The Effect of a Word Prediction Feature on User Performance. *Augmentative and Alternative Communication*, 12(3): 1996, 155-168.
13. Lesher, Gregory and Gerard Rinkus. Domain-specific word prediction for augmentative communication. *Proceedings of the RESNA '02 Annual Conference.*
14. Lesher, Gregory, Bryan Moulton, and Jefferey Higgonbotham. Effects of ngram order and training text size on word prediction. *Proceedings of the RESNA ''99 Annual Conference.*
15. Lesher, Gregory, Bryan Moulton, Jeffery Higginbotham, and Brenna Alsofrom. Limits of human word prediction performance. *Proceedings of California State University Northridge conference, 2002*.
16. Mahajan, Milind, Doug Beeferman, and X. D. Huang. Improved topic-dependent language modeling using information retrieval techniques. *Proceedings. of the International Conference on Acoustics, Speech, and Signal Processing*, 1999.
17. Manning, Christopher and Hinrich Shütze. *Foundations of Statistical Natural Language Processing.* MIT Press, Cambridge MA, 2000.
18. Marcus, Mitchell, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2): 1993.
19. McCoy, Kathleen F. Interface and Language Issues in Intelligent Systems for People with Disabilities. In *Assistive Technology and Artificial Intelligence: Applications in Robotics, User Interfaces and Natural Language Processing*, Vibhu Mittal, Holly Yanco and John Aronis, Editors. Volume 1458, Lecture Notes in AI Series, Springer, 1998.
20. Newell, Alan, John Arnott, Lynda Booth, William Beattie, Bernadette Brophy, and Ian Ricketts. Effect of the "PAL" Word Prediction System on the Quality and quantity of Text Generation. *Augmentative and Alternative Communication*, Volume 8, 1992.
21. Newell, Alan, Stefan Langer and Marianne Kickey. The rôle of natural language processing in alternative and augmentative communication. *Natural Language Engineering,* 4(1): 1998, 1-16.
22. Seymore, Kristie and Ronald Rosenfeld. Using story topics for language model adaptation. *Proceedings of 5th European Conference on Speech Communication and Technology (Eurospeech)*, 1997.
23. Silfverberg, Miika, I. Scott MacKenzie, and Panu Korhonen. Predicting Text Entry Speed on Mobile Phones. *Proceedings of CHI 2000*.
24. Venkatagiri, H. S. Effect of window size on rate of communication in a lexical prediction AAC system. *Augmentative and Alternative Communication*, 10, 1994, 105-112.