Applying Prediction Techniques to Phoneme-based AAC Systems

Keith Vertanen
Department of Computer Science
Montana Tech of the University of Montana
keithv@keithv.com

Ha Trinh, Annalu Waller, Vicki L. Hanson
School of Computing
University of Dundee
{hatrinh, awaller, vlh}@computing.dundee.ac.uk

Per Ola Kristensson
School of Computer Science
University of St Andrews
pok@st-andrews.ac.uk

Abstract

It is well documented that people with severe speech and physical impairments (SSPI) often experience literacy difficulties, which hinder them from effectively using orthographic-based AAC systems for communication. To address this problem, phoneme-based AAC systems have been proposed, which enable users to access a set of spoken phonemes and combine phonemes into speech output. In this paper we investigate how prediction techniques can be applied to improve user performance of such systems. We have developed a phoneme-based prediction system, which supports single phoneme prediction and phoneme-based word prediction using statistical language models generated using a crowdsourced AAC-like corpus. We incorporated our prediction system into a hypothetical 12-key reduced phoneme keyboard. A computational experiment showed that our prediction system led to 56.3% average keystroke savings.

1 Introduction

Over the last forty years there has been an increasing number of high-tech AAC systems developed to provide communication support for individuals with severe speech and physical impairments (SSPI). Most of existing AAC systems can be classified into two categories, namely graphic-based and orthographic-based systems. Graphic-based systems utilize symbols to encode a limited set of frequently used words and utterances, thereby supporting fast access to pre-stored items. However, there is a high cognitive overhead associated with learning the encoding methods of these systems, which can be problematic for many AAC users, especially those with intellectual disabilities. In addition, users of these systems are limited to pre-programmed items rather than being able to create novel words and messages spontaneously. In contrast, orthographic-based AAC systems allow users to spell out their own messages. Prediction techniques, such as character or word prediction, are often applied to improve the usability and accessibility of these systems. However, these systems require users to master literacy skills, a well-documented problem for many children and adults with SSPI (Koppenhaver and Yoder, 1992).

The question arises as to how AAC systems can be designed to enable pre-literate users with SSPI to generate novel words and messages in spontaneous conversations. A potential solution for this question is to adopt a phoneme-to-speech generation approach. This approach allows users to access a limited set of spoken phonemes and blend phonemes into speech output, thereby enabling them to create spontaneous messages without knowledge of orthographic spelling. This approach has been applied in several phoneme-based AAC systems to support communication (Glennen and DeCoste, 1997) and literacy learning (Black et al., 2008). It has also been utilized as an alternative typing method for people with spelling difficulties (Schroeder, 2005).

Despite such potential, phoneme-based AAC systems have been an under-researched topic. In particular, little work has been done on the application of Natural Language Processing (NLP) techniques to these systems. Thus, in this paper we investigate how prediction methods can be incorporated into phoneme-based AAC systems to facil-

NAACL-HLT 2012 Workshop on Speech and Language Processing for Assistive Technologies (SLPAT), pages 19–27,
Montréal, Canada, June 7–8, 2012.

itate phoneme entry. We develop a basic phoneme-based prediction system, which provides predictions at both phoneme and word levels based on statistical language modeling techniques. We use a 6-gram phoneme mixture model and a 3-gram word mixture model trained on a large set of AAC-like data assembled from multiple sources, such as Twitter, Blog, and Usenet data. We take into consideration issues such as pronunciation variants and user accents in the design of our system. We performed a theoretical evaluation of our system on three different test sets using a simulated interface and report results of hit rate and potential keystroke savings. Finally, we propose a number of further studies to extend the current work.

2 Background

2.1 Phoneme-based AAC Systems

The idea of using phonemes in AAC systems was first commercially introduced by Phonic Ear in 1978 in the HandiVoice 110 (Creech, 2004; Glennen and DeCoste, 1997; Williams, 1995). The device provided users with direct access to a mixed vocabulary consisting of pre-programmed words, short phrases, letters, morphemes, and 45 phonemes. Users could generate synthetic speech from phoneme sequences using the Votrax speech synthesizer. Similar to the HandiVoice is the Finger Foniks, a handheld communicator developed by Words+ (Glennen and DeCoste, 1997). The device enables users to access prerecorded messages and a set of 36 phonemes from which they could generate unlimited speech output. Neither of these devices offered any prediction features.

The PhonicStick™, a talking joystick (Black et al., 2008), is a phoneme-based AAC device developed by researchers at the University of Dundee. Unlike the HandiVoice and the Finger Foniks, the primary use of the PhonicStick™ is to facilitate language play and phonics teaching for children with SSPI. The device allows users to access the 42 phonemes used in the Jolly Phonics literacy program (Lloyd, 1998) by moving the joystick along pre-defined paths. A prototype of the PhonicStick™, using a subset of 6 Jolly Phonics' phonemes, has been evaluated with seven children without and with SSPI. Results of the evaluations demonstrated that the participants could create short words using the phonemes. However, some participants with poor hand function experienced significant difficulties in using the joystick to select target phonemes (Black et al., 2008). This suggests that the PhonicStick™ could benefit from prediction mechanisms to reduce the number of difficult joystick movements required for each phoneme entry.

The phoneme-to-speech approach is not only applied in dedicated AAC systems but also in alternative typing interfaces for individuals with spelling difficulties. An example of such applications is the REACH Sound-It-Out Phonetic Keyboard™ (Schroeder, 2005). This on-screen keyboard comprises 40 phonemes and 4 phoneme combinations. It offers two types of prediction features, including phoneme prediction and word prediction. The phoneme prediction feature uses a pronunciation dictionary to determine which phonemes cannot follow the currently selected phonemes. These phonemes are then removed from the keyboard, thereby facilitating users in visually scanning and identifying the next phoneme in the intended word. The word prediction feature also uses a dictionary to search for the most frequently used words that phonetically match the currently selected phoneme sequence. To our knowledge, this is the only currently available system that provides phoneme-based predictions. However, these predictions use a simple dictionary-based prediction algorithm, which does not take into account contextual information (e.g. prior text). There has been little or no published research into how more advanced NLP techniques can be employed to improve the performance of phoneme-based predictions.

2.2 Prediction in AAC Systems

Prediction techniques have been extensively utilized in many AAC systems to achieve keystroke savings and potential communication rate enhancement (Garay-Victoria and Abascal, 2005). There are various prediction strategies that have been developed in these systems, of which the most commonly used are character prediction and word prediction. Character prediction anticipates next probable characters given the preceding characters. It is typically applied in reduced keyboards and scanning-based AAC systems to augment the scanning process (Lesher et al., 1998). Word prediction anticipates the word being entered on the basis of the previously selected characters and

words, thereby saving the user the effort of entering every character of a word.

Most existing prediction systems employ statistical language modelling techniques to perform prediction tasks. Prediction accuracy generally increases with higher-order n-gram language models. However, most systems are limited to 6-gram models for character prediction and 3-gram models for word prediction, as the gain from higher-order models is often small at the cost of considerably increased computational and storage resources. To further improve the prediction performance, a number of advanced language modelling techniques have been investigated, which take into account additional information such as word recency (Swiffin et al., 1987), syntactic information (Hunnicutt and Caarlberger, 2001; Swiffin et al., 1987), semantic information (Li and Hirst, 2005), and topic modelling (Trnka et al., 2006). These techniques have the potential of improved keystroke savings at the cost of increased computational complexity.

A fundamental issue of the statistical-based prediction approach is that its performance is heavily dependent on the size of the training corpus and the degree to which the corpus represents the domain of use. Therefore, in the development of statistical-based prediction for conversational AAC systems, it may be ideal to construct language models from a large corpus of transcribed conversations of real AAC users. However, such a corpus has been unavailable to date. To address this problem, previous research has utilized corpora of telephone transcripts, such as the Switchboard corpus, and performed cleanup processing to make them a more appropriate approximate of AAC communication (Lesher and Rinkus, 2002; Trnka et al., 2006). Vertanen and Kristensson (2011) have recently proposed a novel solution to this problem by creating a large corpus of fictional AAC messages. Using Amazon Mechanical Market, the researchers crowdsourced a small dataset of AAC-like messages, which was then used to select a much larger set of AAC-like data from Twitter, Blog, and Usenet datasets. The language models trained on this AAC-like corpus were proved to outperform other models trained on telephone transcripts (Vertanen and Kristensson, 2011).

3 Phoneme-based Prediction System

Although statistical-based predictions have been a well-studied topic, little or no research has been published on how well these predictions can be adapted to phoneme-based AAC systems. In this section, we describe our phoneme-based prediction system, which employs statistical language modeling techniques to perform phoneme prediction and phoneme-based word prediction. Phoneme prediction predicts probable next phonemes based on the previously entered phonemes. Word prediction predicts the word currently being entered based on the current phoneme prefix and prior words.

3.1 Phoneme Set

Unlike traditional orthographic-based AAC systems that operate on a standard character set, different phoneme-based systems tend to use slightly different phoneme sets. For our prediction system, we use the phoneme set from the Jolly Phonics, a systematic synthetic phonics program widely used in the UK for literacy teaching (Lloyd, 1998). The phoneme set, to be called the PHONICS set, consists of 42 phonemes, with 17 vowels and 25 consonants. By using a literacy-linked phoneme set, our prediction system can readily be integrated into both literacy learning tools (such as the PhonicStick™ joystick (Black et al., 2008)) and communication aids. Other systems that use different phoneme sets can also be easily adapted to utilize our prediction system by providing a phoneme mapping scheme between their phoneme sets and the PHONICS set.

3.2 Pronunciation Dictionary

3.2.1 The PHONICS Dictionary

The development of phoneme-based predictions requires a pronunciation dictionary, which should be accent-specific as pronunciations may vary across different accents. There has been no dictionary to date that contains word pronunciations using the PHONICS set. To address this problem, we built our PHONICS pronunciation dictionary based on the Unisyn[1] lexicon, as it provides facilities for generating dictionaries in different accents. The Unisyn uses the concept of key-symbols (i.e. meta-phonemes) to encode the characteristics of

[1] http://www.cstr.ed.ac.uk/projects/unisyn/

[2] http://aac.unl.edu/vocabulary.html, accessed 4 September

multiple accents into a single base lexicon. Accent-specific rules can then be applied to the base lexicon to produce pronunciations in a given accent.

To create the PHONICS dictionary, we first derived a lexicon in the Edinburgh accent from the base lexicon using a set of Perl scripts supplied with Unisyn. We also performed additional clean-up processing to remove unwanted information, such as stress and boundary markers. We then created a mapping function from the set of 61 phonemes and allophones used in the Edinburgh lexicon to the PHONICS set. As the PHONICS set only contains 42 phonemes, several allophones in the Edinburgh set were mapped to the same phonemes in the PHONICS set. This mapping function was then used to convert the Edinburgh lexicon to the PHONICS pronunciation dictionary. The resulting dictionary consists of 121,004 pronunciation entries for 117,625 unique words.

3.2.2 The Schwa Phoneme

An issue of the phoneme mapping is that the Edinburgh set contains the schwa phoneme (denoted by the symbol '@'), which cannot be mapped to any phonemes in the PHONICS set. The schwa, a reduced form of full vowels in unstressed syllables, occurs in 41,539 entries in the PHONICS dictionary. An example of a word containing the schwa phoneme is *'today' (/t @ d ai/)*. While the schwa is the most commonly used vowel sound in spoken English (Gimson and Cruttenden, 2001), it is not included in the Jolly Phonics teaching as it is a difficult concept to understand for literacy learners at early stages.

The simplest solution for this issue would be to explicitly add the schwa phoneme into the PHONICS set in our prediction system. However, learning to use the schwa correctly can be challenging for users with SSPI and literacy difficulties. Thus, we decided to support two modes in our system, namely the SCHWA_ON and the SCHWA_OFF modes. In the SCHWA_ON mode, the schwa phoneme is explicitly added to the PHONICS set, increasing the set to 43 phonemes. In the SCHWA_OFF mode, the schwa is not added into the PHONICS set and therefore is not offered to the users for selection. To deal with the absence of the schwa, we employed a basic auto-correction method. To search for a word given a phoneme sequence, we apply a limited set of schwa insertion and replacement rules (e.g. replacing vowels with schwas) to generate a set of alternative sequences. These sequences and the original sequence are then used to look up a list of matching words in the PHONICS dictionary. Once the user has selected a word from this list, the correct pronunciation of the selected word (which might include the schwas) would be used to replace the original phoneme sequence in the currently selected phoneme string. This corrected phoneme string would then be input to the phoneme language model (described in Section 3.3.1) to predict probable next phonemes.

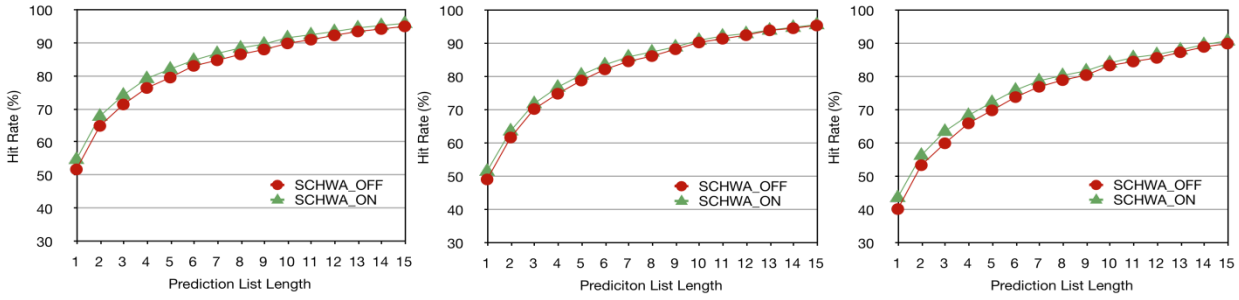
3.3 Phoneme Prediction

We trained a 6-gram phoneme language model starting with training data from:

- Twitter messages collected via the free streaming API between December 2010 and July 2011. 36M sentences, 251M words.
- Blog posts from the ICWSM corpus (Burton et al., 2009). 25M sentences, 387M words.
- Usenet messages (Shaoul and Westbury, 2009). 123M sentences, 1847M words.

We used the crowdsourced data from Vertanen and Kristensson (2011) to select AAC-like sentences using cross-entropy difference selection (Moore and Lewis, 2010). The selection process retained 6.9M, 1.6M, and 2.3M words of data from the Twitter, Blog and Usenet data sets respectively. We converted the words in the selected sentences to pronunciation strings using the PHONICS dictionary. Whenever we encountered a word with multiple pronunciations, we chose a pronunciation at random. If a sentence had a word not in the PHONICS dictionary, we dropped the entire training sentence.

We trained a 6-gram phoneme language model for each of the Twitter, Blog, and Usenet data sets. Estimation of unigrams used Witten-Bell discounting while all higher order n-grams used modified Kneser-Ney discounting with interpolation. We then created a mixture model via linear interpolation with mixture weights optimized on the crowdsourced development set from Vertanen and Kristensson (2011). The optimized mixture weights were: Twitter 0.54, Blog 0.25, and Usenet 0.21. Our final mixture model has 2.0M parameters and a compressed disk size of 14 MB.



(a) SPECIALISTS test set (b) COMM test set (c) SWITCHTEST test set

Figure 1. Hit rates of the phoneme prediction for prediction list lengths 1-15 in the SWCHA_ON and SCHWA_OFF modes. Results on the SPECIALISTS, COMM, and SWITCHTEST test sets.

3.3.1 Hit Rate

We evaluated the accuracy of our phoneme prediction using hit rate. Hit rate (HR) is defined as the percentage of times that the intended phonemes appear in the prediction list:

$$\text{HR} = \frac{\text{Number of times the phoneme is predicted}}{\text{Number of phonemes}} \times 100\%$$

We computed the hit rates for prediction lists of lengths 1-15 in both SCHWA_ON and SCHWA_OFF modes. The results of this evaluation would help inform the decision of the number of predicted items to be presented to the users, which is a key usability factor of prediction systems.

We evaluated the hit rates on the following test sets:

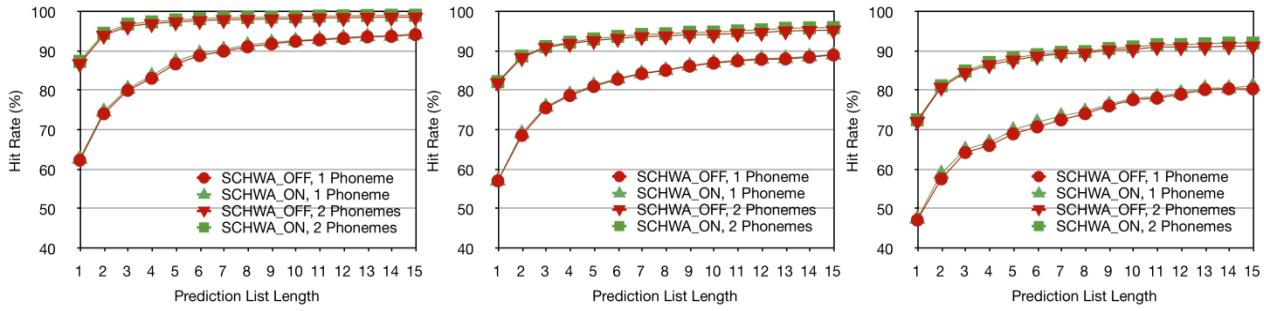
- SPECIALISTS: A collection of context specific conversational phrases recommended by AAC professionals[2]. 966 sentences, 3814 words. Out-of-vocabulary (OOV) rate: 0.05%.
- COMM: A collection of sentences written by college students in response to 10 hypothetical communication situations (Venkatagiri, 1999). 251 sentences, 1789 words. OOV rate: 0.3%.
- SWITCHTEST: Three telephone transcripts taken from the Switchboard corpus, used in Trnka et al. (2009). 59 sentences, 508 words. OOV rate: 0.4%.

These three test sets are used throughout this paper. For each sentence in the test sets, we generated its pronunciation string using the PHONICS dictionary. During this generation, any time we encountered a word with multiple pronunciations, we chose a pronunciation at random. We manually added pronunciations for OOV words. The generated pronunciations were used to calculate the hit rates in the SCHWA_ON mode. We then created a 'non-schwa' version of each pronunciation string, in which we removed all schwa occurrences by either deleting them or replacing them with appropriate vowels in the PHONICS set. The 'non-schwa' pronunciations were used to calculate the hit rates in the SCHWA_OFF mode.

As shown in Figure 1, the hit rate improved as the prediction list length (L) increased in both the SCHWA_OFF and SCHWA_ON modes for all the three test sets. For most L values, the system performed the best on the SPECIALISTS test set and the worst on the SWITCHTEST set. At L=1, the average hit rates for the three test sets were 47.1% in the SCHWA_OFF mode and 50.1% in the SWITCH_ON mode. At L=5 (which is the length usually offered in prediction systems), the average hit rate increased to 76.2% in the SCHWA_OFF mode and 78.4% in the SCHWA_ON mode. At L=15, the system reached high average hit rates of 93.6% in the SCHWA_OFF mode and 94.3% in the SCHWA_ON mode.

The SCHWA_ON mode achieved higher hit rates than the SCHWA_OFF mode for all L values. However, the hit rate differences between these two modes tended to diminish as L increased. At L=1, the average difference for the three test sets was 3.0%. At L=5, the average difference reduced to 2.2%. At L=15, the average difference was very small, at 0.7%.

[2] http://aac.unl.edu/vocabulary.html, accessed 4 September 2011



(a) SPECIALISTS test set (b) COMM test set (c) SWITCHTEST test set

Figure 2. Hit rates of the word prediction for prediction list lengths 1-15 in the SWCHA_ON and SCHWA_OFF modes for 1-phoneme and 2-phoneme prefixes. Results on the SPECIALISTS, COMM, and SWITCHTEST test sets.

3.4 Phoneme-based Word Prediction

We used a publicly available 3-gram word mixture model[3], which was created from three 3-gram models trained on AAC-like data from Twitter, Blog, and Usenet (Vertanen and Kristensson, 2011). Although a 4-gram model trained on the same datasets is also available, it was not used in our system as it has been shown to only slightly outperform the 3-gram model at the cost of a much bigger model size (Vertanen and Kristensson, 2011). Our aim is to keep our prediction system's size reasonably small, thereby allowing it to be easily integrated into devices with limited resources, such as mobile devices.

To perform word prediction given a phoneme prefix, we first search for a set of matching words in the PHONICS dictionary. In the SCHWA_OFF mode, the phoneme prefix is input to the auto-correction function to generate alternative prefixes, which are then used to look up matching words in the dictionary. If there is no matching word, an unknown word (denoted as <unk>) is returned. The matching words are then input to the word model to calculate their probabilities based on up to two prior words.

3.4.1 Hit Rate

We computed the hit rate (HR) of word prediction for prediction list lengths 1-15 in two conditions: (1) after the first phoneme is entered, (2) after the first two phonemes are entered:

$$\text{HR} = \frac{\text{Number of times the word is predicted}}{\text{Number of words}} \times 100\%$$

Figure 2 shows the hit rates of word prediction in the SCHWA_OFF and SCHWA_ON modes on the three test sets. As expected, the hit rates improved as the prediction list length (L) increased. Table 1 summarizes the average hit rates for several list lengths for 1-phoneme and 2-phoneme prefixes. At L=5, the average hit rates were 92.5% in the SCHWA_OFF mode and 93.2% in the SCHWA_ON mode after the first two phonemes are entered. This means that in most cases, the intended word is predicted after two keystrokes. The SCHWA_ON mode achieved higher hit rates than the SCHWA_OFF mode in all cases. However, the hit rate differences between these two modes were very small (<1%), which implies that our auto-correction mechanism was effective.

L	SCHWA_OFF		SCHWA_ON	
	1-phoneme	2-phoneme	1-phoneme	2-phoneme
1	55.6%	80.4%	55.9%	80.8%
5	79.0%	92.5%	79.7%	93.2%
10	86.0%	94.5%	86.2%	95.0%
15	88.0%	95.1%	88.3%	95.8%

Table 1. Average hit rates of word prediction.

4 Theoretical Evaluation

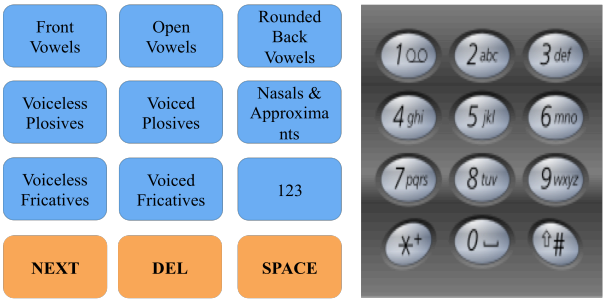
AAC users with physical impairments often experience difficulties in accessing a large number of keys on conventional full-sized keyboards. To address this problem, previous research has proposed the use of reduced keyboards (i.e. keyboards on which each key is assigned a group of charac-

[3] http://www.aactext.org/imagine/lm_mix_top3_3gram_abs0.0.arpa.gz

ters, such as the 12-key mobile phone keyboard) (Arnott and Javed, 1992; Kushler, 1998). Character prediction and word prediction can be applied to these keyboards to disambiguate characters on each key. We adopted this idea by creating a hypothetical 12-key phoneme keyboard and evaluated the benefits of incorporating phoneme prediction and word prediction into the keyboard.

4.1 Phoneme-based Predictive Interface

Our 12-key phoneme keyboard contains 8 phoneme keys, which represent 3 vowel groups and 5 consonant groups. These groups, introduced in the PhonicStick™ talking joystick (Black et al., 2008; Lindström and Peronius, 2010), are formed according to the manner of articulation of the phonemes (see Figure 3a). Each key represents three to seven phonemes; the schwa phoneme is excluded. The phonemes on each key are initially arranged according to the unigram probabilities estimated by our phoneme language model.



a. The 12-key phoneme keyboard

b. The standard 12-key mobile phone keyboard

Figure 3. Phoneme-based reduced keyboard.

The keyboard provides two phoneme entry modes, namely the MULTITAP and the PREDICTIVE modes. In the MULTITAP mode, the user enters a phoneme by pressing a corresponding key repeatedly until the intended phoneme appears (e.g. pressing the 'Unvoiced Plosives' key 3 times to enter /p/). In the PREDICTIVE mode, the keyboard utilizes our prediction system in its SCHWA_OFF mode to predict probable next phonemes and words. Each time the user presses a key the phoneme prediction is applied to guess which of the possible phonemes on the pressed key is actually the user's intended phoneme. If the prediction is incorrect, the user can repeatedly press the NEXT key until the correct phoneme is selected. After each phoneme selection, we present a list of up to 5 predicted words. We only offer word predictions after the first phoneme of a new word is entered. If the intended word appears in the prediction list, we assume it takes one keystroke for the user to add the word and a following space to the current sentence (this can be implemented using automatic scanning (Glennen and DeCoste, 1997)).

4.2 Results

We evaluated our prediction system using two commonly used metrics: keystroke savings and keystrokes per character.

4.2.1 Keystroke Savings

Keystroke Savings (KS) is defined as the percentage of keystrokes that the user saves by using prediction methods compared to using the MULTITAP method:

$$KS = \left(1 - \frac{\text{Keystrokes}_{\text{PREDICTION}}}{\text{Keystrokes}_{\text{MULTITAP}}}\right) \times 100\%$$

We computed KS on the three test sets for three methods: (1) only phoneme prediction (PP), (2) only word prediction (WP), (3) combined phoneme prediction and word prediction (PP+WP) (i.e. the PREDICTIVE mode).

As shown in Figure 4, a combined phoneme and word prediction method performed the best with an average keystroke savings of 56.3%. Using only word prediction led to a 46.4% average KS while using only phoneme prediction resulted in 29.9% average KS.

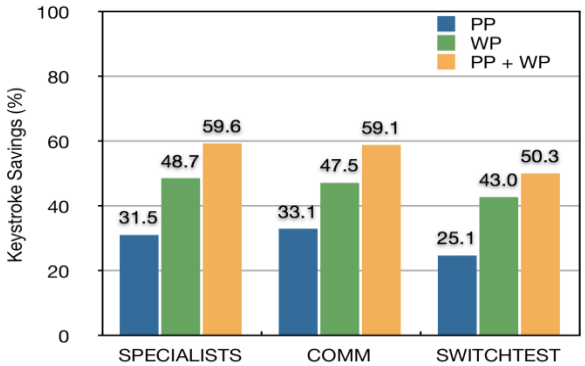


Figure 4. Keystroke Savings (KS) for prediction methods on three test sets.

4.2.2 Keystrokes Per Character

Keystrokes per character (KSPC) is defined as the average number of keystrokes required to produce a character in the test set:

$$KSPC = \frac{\text{Keystrokes}}{\text{Number of characters (including spaces)}}$$

The evaluation of KSPC allows us to compare our keyboard with existing character-based reduced keyboards. We computed the KSPC for four methods: (1) MULTITAP, (2) PP, (3) WP, (4) PP+WP. For comparison, we also calculated the KSPC for a standard 12-key mobile phone alphabetic keyboard (Figure 3b), which uses the character-based multitap method for text entry.

As shown in Figure 5, our frequency-based phoneme keyboard outperformed the standard mobile phone keyboard even when no prediction methods are applied (i.e. in the MULTITAP mode) (see Figure 5). At an average KSPC of 1.568, our keyboard required 19.1% fewer keystrokes per character than the mobile phone multitap keyboard (KSPC=1.937). There are two reasons that might explain this result. First, on average one phoneme represents more than one character (in our dictionary, the character/phoneme ratio is 1.208). Second, our keyboard's phonemes were initially ordered by the unigram frequencies.

When applying only phoneme prediction, the average KSPC decreased to 1.100, which closely approaches the KSPC of a QWERTY keyboard (KSPC=1). The KSPC further reduced to 0.841 with solely word prediction and 0.685 with combined phoneme and word prediction.

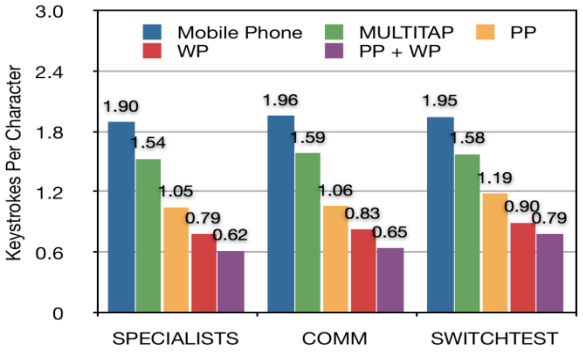


Figure 5. Keystrokes Per Character (KSPC) for different text entry methods on three test sets.

5 Conclusions and Future Work

In this paper we have described how statistical language modeling techniques can be used to provide phoneme prediction and word prediction for phoneme-based AAC systems. Using hit rate measurement we demonstrated how the prediction accuracy improved as the prediction list length increased. However, a large prediction list might result in an increased time and cognitive workload required from the user to scan the list and select the desired item. Therefore, hit rate data need to be combined with empirical experiments with real users in order to determine an appropriate prediction list length.

We evaluated our prediction system on a 12-key phoneme keyboard, in which phonemes are grouped based on the manner of articulation and ordered using our phoneme unigram frequencies. We showed that we could achieve a potential keystroke savings of 56.3% by applying a combined phoneme and word prediction to our keyboard. Using word prediction alone proved to be more effective than using phoneme prediction alone, in terms of keystroke savings.

We plan to take this work forward by exploring two complementary research directions.

First, we plan to conduct empirical experiments with a group of AAC users to evaluate the usability of our phoneme predictive keyboard. We are interested in finding out if the potential keystroke savings can be translated into an actual keystroke savings and communication rate enhancement. In addition, we will analyze user's errors in phoneme selection, which can be used to produce a more advanced auto-correction method.

Second, we will explore how our prediction system can be integrated into existing phoneme-based AAC systems rather than our reduced keyboard. In particular, we will focus on the REACH Sound-It-Out Phonetic Keyboard™ (Schroeder, 2005), which uses a different phoneme set than our PHONICS set, and the PhonicStick™ (Black et al., 2008), which has the same phoneme groupings as our keyboard.

Finally, we will investigate how NLP techniques, such as the joint-multigram model (Bisani and Ney, 2008), can be applied to automatically generate orthographic spellings for OOV words. Our current system simply uses a <unk> placeholder for OOV words. While these words can still be spoken out by synthesizing their phoneme strings, it is potentially more beneficial to suggest actual spellings than to use such a placeholder.

References

John L. Arnott and Muhammad Y. Javed (1992). Probabilistic character disambiguation for reduced keyboard using small text samples. *Augmentative and Alternative Communication, 8*(3), 215-223.

Maximilian Bisani and Hermann Ney (2008). Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication, 50*(5), 434-451.

Rolf Black, Annalu Waller, Graham Pullin, and Eric Abel (2008). *Introducing the PhonicStick: Preliminary evaluation with seven children.* Paper presented at the 13th Biennial Conference of the International Society for Augmentative and Alternative Communication Montreal, Canada.

Kevin Burton, Akashay Java, and Ian Soboroff (2009). *The ICWSM 2009 Spinn3r dataset.* Paper presented at the 3rd Annual Conference on Weblog and Social Media.

Rick Creech (2004). Rick Creech, 2004 Edwin and Esther Prentke AAC Distinguished Lecturer, from http://www.aacinstitute.org/Resources/PrentkeLecture/2004/RickCreech.html

Nestor Garay-Victoria and Julio Abascal (2005). Text prediction systems: a survey. *Universal Access in the Information Society, 4*, 188-203.

Alfred. C. Gimson and Alan Cruttenden (2001). *Gimson's Pronunciation of English*: Hodder Arnold.

Sharon L. Glennen and Denise C. DeCoste (1997). *The Handbook of Augmentative and Alternative Communication*: Thomson Delmar Learning.

Sheri Hunnicutt and Johan Caarlberger (2001). Improving Word prediction using markov models and heuristic methods. *Augmentative and Alternative Communication, 17*(4), 255-264.

David A. Koppenhaver and David E. Yoder, D (1992). Literacy issues in persons with severe speech and physical impairments. In R. Gaylord-Ross, Ed. (Ed.), *Issues and research in special education* (Vol. 2, pp. 156-201). NY: Teachers College Press, Columbia University, New York.

Cliff Kushler (1998). *AAC: Using a reduced keyboard.* Paper presented at the Technology and Persons with Disabilities Conference, Los Angeles, USA.

Gregory W. Lesher and Gerald J. Rinkus (2002). *Domain-specific word prediction for augmentatve communication.* Paper presented at the The RESNA '02 Annual Conference.

Gregory W. Lesher, Bryan J. Moulton, and Jeffrey D. Higginbotham (1998). Techniques for augmenting scanning communication. *Augmentative and Alternative Communication, 14*, 81-101.

Jianhua Li and Graeme Hirst (2005). *Semantic knowledge in word completion*. Paper presented at the 7th International ACM SIGACCESS Conference on Computers and Accessibility.

Nina Lindström and Irmeli Peronius (2010). *The PhonicStick nursery study: Can phonological awareness be initiated by using a speaking joystick.* Uppsala University.

Susan M. Lloyd (1998). *The Phonics Handbook*. Chigwell: Jolly Learning Ltd.

Robert C. Moore and William Lewis (2010). *Intelligent selection of language model training data.* Paper presented at the 48th Annual Meeting of the Association of Computational Linguistics.

James E. Schroeder (2005). *Improved spelling for persons with learning disabilities.* Paper presented at the The 20th Annual International Conference on Technology and Persons with Disabilities, California, USA.

Cyrus Shaoul and Chris Westbury (2009). A USENET corpus (2005-2009). University of Alberta, Canada.

Andrew L. Swiffin, John L. Arnott, and Alan Newell (1987). *The use of syntax in a predictive communication aid for the physically handicapped.* Paper presented at the RESNA 10th Annual Conference, San Jose, California.

Andrew L. Swiffin, John L. Arnott, Andrian J. Pickering, and Alan Newell (1987). Adaptive and predictive techniques in a communication prosthesis. *Augmentative and Alternative Communication, 3*(4), 181-191.

Keith Trnka, Debra Yarrington, Kathleen F. McCoy, and Christopher Pennington (2006). *Topic modeling in fringe word prediction for AAC*. Paper presented at the 11th International Conference on Intelligent User Interfaces.

Keith Trnka, John McCaw, Debra Yarrington, Kathleen F. McCoy, Christopher Pennington (2009). User interaction with word prediction: The effects of prediction quality. *ACM Transactions on Accessible Computing, 1*(3), 1-34.

Horabail S. Venkatagiri (1999). Efficient keyboard layouts for sequential access in augmentative and alternative communication. *Augmentative and Alternative Communication, 15*(2), 126-134.

Keith Vertanen and Per Ola Kristensson (2011). *The imagination of Crowds: Conversational AAC language modelling using crowdsourcing and large data sources.* Paper presented at the International Conference on Empirical Methods in Natural Language Processing (EMNLP), Edinburgh, United Kingdom.

Michael B. Williams (1995). *Transitions and transformations.* Paper presented at the 9th Annual Minspeak Conference, Wooster, OH.