

Dunlop, M.D. and Masters, M. Montgomery (2009) Pickup usability dominates: a brief history of mobile text entry research and adoption. *International Journal of Mobile Human Computer Interaction* , 1 (1). pp. 42-59. ISSN 1942-390X

<http://strathprints.strath.ac.uk/16843/>

Strathprints is designed to allow users to access the research output of the University of Strathclyde. Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. You may not engage in further distribution of the material for any profitmaking activities or any commercial gain. You may freely distribute both the url (<http://strathprints.strath.ac.uk>) and the content of this paper for research or study, educational, or not-for-profit purposes without prior permission or charge. You may freely distribute the url (<http://strathprints.strath.ac.uk>) of the Strathprints website.

Any correspondence concerning this service should be sent to The Strathprints Administrator: [eprints@cis.strath.ac.uk](mailto:eprints@cis.strath.ac.uk)

# Pickup Usability Dominates: a brief history of mobile text entry research and adoption

Mark David Dunlop and Michelle Montgomery Masters

University of Strathclyde, UK

---

## Abstract

*Text entry on mobile devices (e.g. phones and PDAs) has been a research challenge since devices shrank below laptop size: mobile devices are simply too small to have a traditional full-size keyboard. There has been a profusion of research into text entry techniques for smaller keyboards and touch screens: some of which have become mainstream, while others have not lived up to early expectations. As the mobile phone industry moves to mainstream touch screen interaction we will review the range of input techniques for mobiles, together with evaluations that have taken place to assess their validity: from theoretical modelling through to formal usability experiments. We also report initial results on iPhone text entry speed.*

**Keywords:** *Text entry, Ambiguous keyboards, Unambiguous keyboards, Touch-screen text entry, Handwriting recognition, Evaluation, Usability techniques, Predictive text entry, User studies*

---

Many mobile services such as text/instant messaging, email, web searching and diary operations require users to be able to enter text on a phone. Text messaging has even overtaken voice calling as the dominant use of mobile phones for many users with mobile email rapidly spreading. Handheld screen technologies are also making it increasingly convenient to read complex messages or documents on handhelds, and cellular data network speeds are now often in excess of traditional wired modems and considerably higher in wi-fi hotspots. These technological developments are leading to increased pressure from users to be able to author complex messages and small documents on their handhelds. Researchers in academia and industry have been working since the emergence of handheld technologies for new text entry methods that are small and fast but easy-to-use, particularly for novice users. This paper will look at different approaches to keyboards, different approaches to stylus-based entry, and how these approaches have been evaluated to establish which techniques are actually faster or less error-prone. The focus of the paper is both to give a perspective on the breadth of research in text entry and also to look at how researchers have evaluated their work. Finally, we will look at perceived future directions attempting to learn from the successes and failures of text entry research. Throughout this paper we will cite words-per-minute (wpm) as a fairly standard

measure of typing speed, for reference highly skilled office QWERTY touch typists achieve speeds of around 135wpm while hand-writing with pen and paper achieves only about 15wpm.

## KEYBOARDS

The simplest and most common form of text entry on small devices, as with large devices, is a keyboard. Several small keyboard layouts have been researched that try to balance small size against usability and text entry speed. Keyboards can be categorized as unambiguous, where one key-press unambiguously relates to one character, or ambiguous, where each key is related to many letters (e.g. the standard 12-key phone pad layout where, say, 2 is mapped to *ABC*). Ambiguous keyboards rely on a disambiguation method, which can be manually driven by the user or semi-automatic with software support and user correction. This section looks first at unambiguous mobile keyboard designs, then at ambiguous designs and, finally, discusses approaches to disambiguation for ambiguous keyboards.

### Unambiguous Keyboards

Small physical keyboards have been used in mobile devices from their very early days on devices such as the Psion Organiser in 1984 and the Sharp Wizard in 1989 and have seen a recent resurgence in devices targeting email users, such as most of RIM's Blackberry range. While early

devices tended to have an alphabetic layout, the standard desktop QWERTY family of layouts, e.g. QWERTY, AZERTY, QWERTZ and QZERTY, was soon adopted as there is strong evidence that alphabetic layouts give no benefits even for novice users (Norman, 2002; Norman & Fisher, 1982). When well-designed, small QWERTY keyboards can make text entry fast by giving the users good physical targets and feedback with speeds measured in excess of 60wpm (Clarkson et al 2005). However, there is a strong design trade-off between keys being large enough for fast, easy typing and overall device size with large-fingered users often finding the keys simply too small to tap individually at speed. Physical keyboards also interact poorly with touch-screens, where one hand often needs to hold a stylus, and they reduce the space available on the device for the screen.

The QWERTY keyboard layout was designed as a compromise between speed and physical characteristics of traditional manual typewriters: the layout separates commonly occurring pairs of letters to avoid head clashes on manual typewriters and is imbalanced between left and right hands. Faster touch-typing office keyboards such as the Dvorak keyboard (Fig 1) are significantly faster but have not been widely adopted – primarily because of the learning time and invested skill-set in QWERTY keyboards. This investment has been shown to carry over into smaller devices, where the sub-optimality issue is even stronger as users tend to type with one or two thumbs – not nine fingers envisaged of touch-typists. While optimal mobile layouts could be designed around two-thumb entry, these are likely to be so different from users' experiences that initial use would be very slow and, as with the Dvorak, rejected by end users (and would still be sub-optimal for one-thumb use!).



Figure 1: Dvorak keyboard

The half-QWERTY mobile keyboard (Matias, MacKenzie, & Buxton, 1996) (fig 2-left) builds on QWERTY skills and the imbalance between left and right hands by halving the keyboard in the centre. The keyboard has a standard left half of a QWERTY keyboard, while the user holds the space bar to *flip* the keyboard to give the right side letters. Targeting smaller size and fast one-handed entry, experiments have shown that users of the half-QWERTY keyboard quickly achieved consistent speeds of 30 words per minute or higher (when using a keyboard with

desktop-sized keys). The FrogPad™ is a variant using an optimised keyboard, so that use of the “right side” of the keyboard is minimised (fig 2-right). (Matias et al., 1996) predicted an optimised pad would lead to a speed increase of around 18% over the half-qwerty design. FrogPad™ Inc now manufacture an optimised keyboard along these lines and claim 40+ words per minute typing speeds.



Fig 2: Simplified Half-QWERTY and FrogPad™

Neither half-keyboards have yet to be integrated into mobiles, while the FastTap™ keyboard, however, has been targeted at mobile devices from initial conception. This patented technology takes a different approach to miniaturisation by including an alphabetic keyboard as raised keys between the standard numeric keys of a phone pad – giving direct non-ambiguous text entry on a very small platform while preserving the standard 12-key keypad currently used by over 90% of mobile users globally (see figure 3). Experiments (Cockburn & Siresena, 2003) have shown that FastTap™ is considerably faster and easier to use for novice users than more standard predictive text approaches and the two approaches perform similarly for expert users (once practiced, FastTap users in their trial achieved 9.3wpm with T9™ users achieving 10.8wpm – somewhat slower than in other trials, see below for discussion of predictive text and T9).



Fig 3: FastTap™ phone keyboard

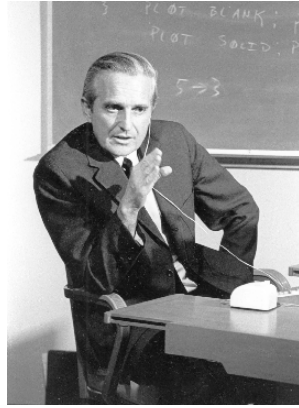


Fig 4: Sample chord keyboards (Douglas Engelbart and Microwriter Agenda)

A drastically different unambiguous keyboard approach is to use chords – multiple simultaneous key-presses mapping to a single character, either using one or both hands. Chord keyboards can give extremely fast entry rates, with court stenographers reaching around 225wpm using a two-handed chord keyboards. One-handed chord keyboards are by definition palm-sized and were originally envisaged as the ideal partner to the mouse (Engelbart & English, 1968), allowing users to enter text and point at the same time. Single-handed chord keyboards have been used in mobile devices (fig 4 right, shows the Agenda organiser including an alphabetic keyboard surrounded by a chord keyboard). However, the learning time is prohibitive with few users willing to learn the chords required to use these keyboards. Furthermore, the keyboards are not usable without training – users cannot guess how to use them when first picking up a device. Thus, despite size and speed advantages, chord keyboards are generally considered too alien for main-stream devices and rarely appear on consumer products.

### Ambiguous keyboards

The most common ambiguous keyboard, and the dominant keyboard for mobile phones, is still the telephony ISO/IEC standard 12-key phone keypad (e.g. fig 5 left). Originally envisaged for name-based dialling of telephone area codes, this keyboard is labelled with groups of three or four letters on each of the physical keys 2 through 9 plus numeric digits (with the 1, \*, # and 0 keys typically acting as space, shift and other control keys depending on handset). These can, however, represent many characters once accented characters are included (e.g. 2 maps to ABC2ÄÆÅÄÄÄÇ). The method of disambiguating the multiple letters per key is discussed later. Recently some phones have been released with a slightly stretched mobile phone pad, typically with two extra columns, reducing the number of characters per key and considerably improving disambiguation accuracy (fig 5 right).



Fig 5: 12-key phone pad (Nokia N73) and stretched phone pad (BlackBerry™ 7100)

While the 12-key mobile phone pad is the smallest commonly found keyboard layout, there has been some work on very small keyboards/devices with as few as three keys. Text entry on these usually involves cursor movement through the alphabet using the 3-key *date-stamp* method widely used in video games (left and right scroll through an alphabetic strip of letters with fire entering the current letter) and 5-key variant using a joystick with a 2D keyboard display. Experiments have shown users of 3-key date-stamp entry can achieve around 9wpm while with a 5-key QWERTY layout users can reach around 10-15wpm (Bellman & MacKenzie, 1998; MacKenzie, 2002b). These experiments also tried dynamically adjusting the layout based on probabilities of next letter but these didn't have the expected speed-up, probably due to the extra attention load of users cancelling out the reduced time to select a letter. An alternative approach for non-ambiguous very small keyboard entry is to use short-codes representing the letters, for example short sequences of cursor keys. (Evreinova, Evreino, & Raisamo, 2004) showed that users could achieve good entry speeds with 3-key combinations of cursor keys, e.g. left-up-left for A, and that, despite high initial error rates, users could learn the codes quickly.

### Disambiguation

The traditional approach to disambiguating text entry on a mobile phone keypad is the manual *multi-tap* approach: users press keys repeatedly to achieve the letter they want, e.g. on a standard phone keypad 2 translates to A with 22 translating to B etc. This approach has also been adopted in many other domestic devices such as video remote

controls. Multi-tap leads to more key-strokes than an unambiguous keyboard, as users have to repeatedly click for most letters, and to a problem with disambiguating a sequence of letters on the same key, e.g. CAB is 222222. Users typically manually disambiguate this by either waiting for a timeout between subsequent letters on the same key or hitting a *time-out kill* button (e.g. right cursor key) – clearly an error-prone process and one that slows users down. (Wigdor & Balakrishnan, 2004) refer to multitap as an example of *consecutive* disambiguation – the user effectively enters a key then disambiguates it. An alternative manual disambiguation approach is *concurrent* disambiguation; here users use an alternative input method, e.g. tilting the phone (Wigdor & Balakrishnan, 2003) or a small chord-keyboard on the rear of the phone (Wigdor & Balakrishnan, 2004), to disambiguate the letter as it is entered. While clearly potentially much faster than multi-tap and relatively easy to use, this approach has not yet been picked up by device manufacturers.

Aimed at overcoming the problems of multi-tap, predictive text entry approaches use language modelling to map from ambiguous codes to words so that users need only press each key once, for example mapping the key sequence 4663 directly to *good*. While there are clearly cases where there are more than one match to the numeric key sequence (e.g. 4663 also maps to *home* and *gone* amongst others), these are surprisingly rare for common words. The problem of multiple matches can be alleviated to a large extent by giving the most likely word as the first suggestion then allowing users to scroll through alternatives for less likely words. Based on a dictionary of words and their frequency of use in the language, users get the right word suggested first around 95% of the time (Gong & Tarasewich, 2005). AOL-Tegic's T9 (Grover, King, & Kushler, 1998; Kushler, 1998) industry-standard entry method is based around this approach and is now deployed on over 2 billion handsets. Controlled experiments have shown this form of text entry considerably out-performs multi-tap (Dunlop & Crossan, 2000; James & Reischel, 2001), with text entry rising from around 8wpm for multitap to around 20 for T9. While predictive text entry is very high quality, it is not perfect and can lead to superficially unrelated predictions that are undetected as users tend to type without monitoring the screen (e.g. a classic T9 error is sending the message *call me when you are good* rather than *are home*). The main problem, however, with any word prediction system is handling out-of-vocabulary words – words that are not known to the dictionary cannot be entered using this form of text entry. The usual solution is

to force users into an “add word” dialogue where the new word is entered in a special window using multi-tap – clearly at a considerable loss of flow to their interaction and reduction in entry speed. As most people do not frequently enter new words or place/people names, this is not a major long-term problem. However, it does considerably impact on initial use and can put users off predictive text messaging as they constantly have to teach new words to the dictionary in the early days of using a new device. This in turn impacts on consumer adoption with many people not using predictive text despite it clearly being faster for experienced users.

An alternative approach to dictionary and word-level disambiguation is to use letter-by-letter disambiguation where letters are suggested based on their likelihood given letters already entered in the given word or likely letters at the start of a word (e.g. in the clearest case in English, a *q* is most likely to be followed by a *u*). This gives the user freedom to enter words that are not in the dictionary and considerably reduces the memory load of the text entry system (no longer an issue with phones but still an issue on some devices). Experiments using this approach (MacKenzie, Kober, Smith, Jones, & Skepner, 2001) showed key-strokes halved and speed increased by around 36% compared with multi-tap. They also claim that this speed is inline with T9 entry and that their approach out-performs T9 by around 30% when as few as 15% of the least common words are missing from the predictive dictionary. Predicting letters based on previous letters is actually a specific implementation of Shannon’s approach to prediction based on n-grams of letters (Shannon, 1951). Some work has been carried out to extend this to the word-level and shows good promise: for example bi-gram word prediction in Swedish with word completion reduced keystrokes by between 7 and 13% when compared with T9 (Hasselgren, Montnemery, Nugues, & Svensson, 2003).

In work on watch-top text entry we (Dunlop, 2004) found that moving to a 5-key pad reduced accuracy from around 96% to around 81% with approximately 40% reduction in text entry speed – however, still considerably faster than 2D date-stamp approaches for very small devices. An interesting alternative input method for small devices is to use a touch-wheel interfaces, such as those on iPods™. (Proschowsky, Schultz, & Jacobsen, 2006) developed a method where users are presented with the alphabet in a circle with a predictive algorithm increasing the target area for letters based on the probability of them being selected next, so that users are more likely to hit the correct target when tapping on the touch-wheel. User trials showed around 6-7 words per minute entry rates for novices, about 30% faster than the

same users using a 1D date-stamp approach on a touch-wheel.

The letters on an ambiguous phone keypad do not, of course, need to be laid out alphabetically. Here the disambiguation method introduces an additional aspect to designing an optimal layout: the letters can be rearranged to minimise the level of ambiguity for a given language in addition to looking at minimising finger movement. However, experiments predict that a fully-optimised phone layout would improve text entry rates by only around 3% for English (Gong & Tarasewich, 2005). We found a larger but still small improvement of around 8% in keystrokes for a pseudo-optimised 4-key letter layout (Dunlop, 2004). Gong & Tarasewich do, however, show that stretching the standard phone pad from eight to twelve keys for text entry (see fig 5) is likely to reduce prediction errors by around 65% for optimised keyboard layouts (Gong & Tarasewich, 2005).

## TOUCH-SCREEN TEXT ENTRY

Compared to mobile phones, personal organisers (PDAs) have made more use of touch screens and stylus interaction as the basis of interaction and this is now emerging on high-end phones such as Apple’s iPhone. This frees up most of the device for the screen and leads to natural mouse-like interaction with applications. Lack of a physical keyboard has led to many different approaches for text entry on touch-sensitive screens that will be discussed in this section: on-screen (or soft) keyboards, hand-writing recognition and more dynamic gesture-based approaches.



Fig 6: iPhone™ on-screen QWERTY keyboard

### On-screen keyboards

A simple solution to text entry on touch-screens is to present the user with an on-screen keyboard that the user can tap on with a stylus, or on larger touch-screens with their fingers. The most common implementation is to copy the QWERTY layout onto a small touch sensitive area at the bottom of the screen (fig 6). Following a similar experimental protocol to (James & Reischel, 2001) we

conducted an initial experiment on three expert iPhone users. James & Reischel measured expert performance on chat style messages at 26wpm; using the same phrases our initial study showed iPhone entry around 50wpm (mean 51.6, stdev 1.5 with similar error rates, though the iPhone spell checker corrected about half of these). Although a very small independent sample study, this does indicate that practiced iPhone users may be up to twice as fast as T9 users.

As with physical keyboards and keypads, there has been research into better arrangements of the keys for touch-screens. Mackenzie and his team have conducted a series of experiments on alternative layouts that are optimised for entry using a stylus (single touch entry). They investigated both unambiguous keyboards and an optimised 12-key ambiguous keypad, inspired by the success of T9 and the fundamental rule of interaction that large targets are faster to tap than small ones (Fitts, 1954). Their results estimate that an expert user could achieve 40+ wpm on a soft QWERTY keyboard with novice soft-keyboard users achieving around 20 wpm (MacKenzie, Zhang, & Soukoreff, 1999). The alternative layouts were predicted to give higher entry rates for expert use: the unambiguous Fitaly layout was predicted to reach up to 56wpm and ambiguous JustType 44wpm (Fig 7). However, novice users achieved only around 8wpm using these alternative keyboard layouts – highlighting the carry-over effect of desktop QWERTY layout.

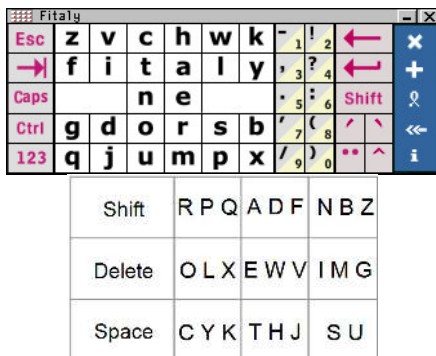


Fig 7: Fitaly and JustType keyboard layouts

While simple and fast, the on-screen keyboard approach can be tiring for users as they are required to repeatedly hit very small areas of the screen. The patented technology underlying the XT9™ Mobile Interface from Tegic Communications attempts to address this problem by including a level of disambiguation in an otherwise unambiguous keyboard (Robinson & Longe, 2000). For example, if the user taps letters adjacent to the letters in the intended word, then the more likely letters are used instead of the letters actually tapped. Their approach defaults to the most likely full word given the approximate letters

entered, while offering alternative corrections and word completions as well as the letters actually typed (fig 8). XT9 technologies have been developed by Tegic for multiple platforms, including hand-printing and small physical keyboards.

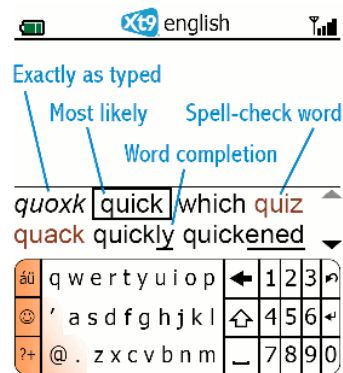


Fig 8: Sample XT9™ Mobile Interface

## Handwriting

To many the obvious solution to text entry on handheld devices is handwriting recognition. Modern hand-writing recognition systems, for example on Windows™ Vista™ tablets, are extremely good at recognising in-dictionary words but struggle on words that are not previously known and are inherently limited by writing speeds (about 15 wpm (S. Card, Moran, & Newell, 1983)). Furthermore, handwriting recognition needs a reasonably large physical space people to write in and processing power that is more in line with modern laptops/tablets than phones. To target mobiles better, the unistroke (Goldberg & Richardson, 1993) approach introduced a simplified alphabet to reduce both the processing complexity and the space, and for experienced users the time, needed for writing while increasing accuracy. Here each letter is represented as a single stroke with letters typically drawn on top of each other in a one letter wide slot and requires users to learn a new alphabet (Fig 9). Palm popularised a more intuitive version, Graffiti™, on their palmtops – a mostly unistroke alphabet, Graffiti™ was composed mainly of strokes with high similarity to standard capital letters. CIC's Jot™ alphabet provides a mix of unistroke and multistroke letters and is deployed on a wide range of handhelds. Experiments comparing hand-printing with other text entry methods are rare, but a comparison between hand-printing, QWERTY-tapping and ABC-tapping on pen-based devices (MacKenzie, Nonnecke, McQueen, Riddersma, & Meltz, 1994) showed that a standard QWERTY layout can achieve around 23wpm while hand-printing achieved only 17wpm and alphabetic soft-keyboard only 13wpm.

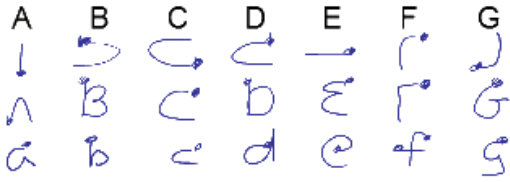


Fig 9: Unistroke, Graffiti™ and Jot™ sample letters

### Word suggestion

Word completion and suggestion can also be used to help users by allowing them to pick full words without entering all the letters. This was first popularised with CIC's WordComplete™ (figure 10 left), which suggested common word completions for partially entered words. Similar technologies are used on the eZiType™ and XT9 technologies deployed on some mobile phones. While tempting, word suggestion and word completion needs to perform very well in order to give users a real benefit – savings in terms of letters entered can be dominated by extra time reading and reacting to on-screen suggestions. We (Dunlop & Crossan, 2000) estimated that simple word completion would reduce keystrokes by 17% but our model-based evaluation (see section 4.1) predicted an approximate halving of entry speed once user interruption time was taken into account. Some recent advances, however, have shown that when based on more complex language models word suggestion and completion can be beneficial with novice users increasing typing speed by around 35% using Apaptxt™ suggestions on a soft keyboard (Dunlop, Glen, Motaparti, & Patel, 2006). Apaptxt language models also self-tune over time by learning patterns of use in the user's language to further improve suggestions for the individual user and his/her context of use (fig 10 centre and right shows off the shelf suggestions and those after repeating a single phrase).

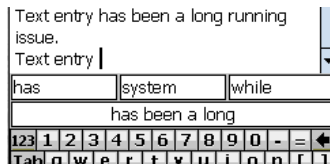
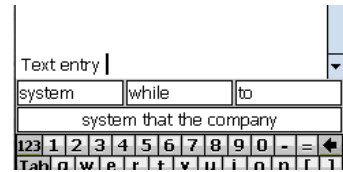
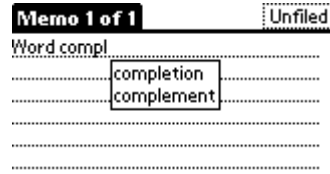


Fig 10: WordComplete™ (left) and Apaptxt™ (centre and right)

### Gesture-based input

Gesture-based interaction attempts to combine the best of visual keyboards with easy-to-remember stylus movements to gain faster and smoother, while still easy-to-learn, text entry. Building on our motor-memory for paths, approaches such as Cirrin (Mankoff & Abowd, 1998), Quikwriting (Perlin, 1998) and Hex (Williamson & Murray-Smith, 2005) are based on the user following a path *through* the letters of the word being entered (Fig 11). For on-screen approaches this achieves faster entry rates than single character printing with reduced stress and fatigue when writing. Furthermore, in the case of Hex, the approach can be used in one-handed on devices with accelerometers/tilt sensors.

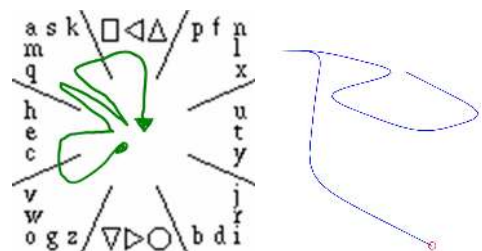


Fig 11: Quikwriting (left) and the Hex entry for "was" (omitting letter display)(right)

Gestures can be combined with more conventional soft keyboards so that users can choose to tap individual letters, improving pick-up-and-use usability, or to enter words in one gesture by following the path of the letters on the touch keyboard (experts can then enter the gestures anywhere on screen) (Zhai & Kristensson, 2003).



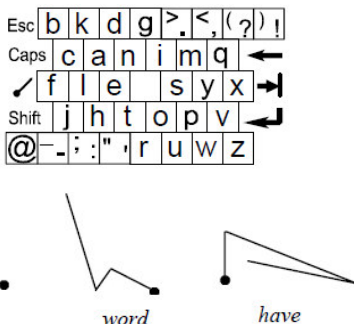


Fig 12: The ATOMIK keyboard with SHARK shortcuts

Dasher is a drastically different approach to text entry that attempts to exploit interactive displays more than traditional text entry approaches. In Dasher (Ward, Blackwell, & MacKay, 2000) (fig 13), letters scroll towards the user and (s)he picks them by moving the stylus up and down as the letters pass the stylus. The speed of scrolling is controlled by the user moving the stylus left and right with predictive text entry approaches dynamically changing the space allocated to each letter (so that likely next letters are given more space than less likely ones, but all letters are available at each stage). Experiments show that users can enter at over 30 words per minute.

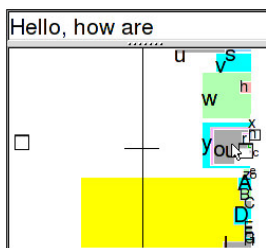


Fig 13: Dasher

## EVALUATION

Unlike many areas of mobile technology where market forces and commercial ingenuity dominate, the field of text entry has benefited from considerable scientific study to establish the benefits of one method over another. These studies have been conducted by academic and industrial research groups, often in collaboration, and are used both to compare techniques and to tune their usage to how users actually enter text. Much of the related evaluation work and results have already been discussed above, in this section we focus on the evaluation methods themselves.

### Technical evaluation

The literature commonly uses three methods for reporting the performance of text entry: *average ranked list position* (ARP), *disambiguation*

*accuracy* (DA) and *keystrokes per character* (KSPC).

The average ranked-list position (e.g. (Dunlop & Crossan, 2000)) for evaluating ambiguous text entry methods is calculated in two phases. First language models, e.g. in the simplest case word frequencies, are learned from a corpus appropriate to the target language. Once trained, the second phase involves processing the same corpus one word at a time. Each word taken from the corpus is encoded using the ambiguous key-coding for the target keypad (e.g. *home* is encoded as 4663) and a ranked list of suggested words produced for that encoding based on the learned language model. The position of the target word in this list is averaged over all words to give the average ranked-list position for that corpus and keypad. An ARP value of 1.0 indicates that the correct word was always in the first position in the ranked list of suggestions, a value of 2.0 that, on average, the correct word was second in the ranked list. We predicted an ARP value of around 1.03 for a large corpus of English language newspaper articles using a standard phone keypad layout. ARP naturally biases the averaging process so that words are taken into account proportionally to their occurrence in the text corpus.

Disambiguation accuracy (e.g. (Gong & Tarasewich, 2005)) reports the percentage of times the first word suggested by the disambiguation process is the word the user intended – a DA value of 100% implies the disambiguation process always give the correct word first, while 50% indicates that it only manages to give the correct word first half of the time. Gong and Tarasewich reported DA of 97% for written English corpus and 92% for SMS messages (both on a phone pad). This is a more intuitive and direct measure than ARP, but does not take into account the performance of words that do not come first in the list.

KSPC (MacKenzie, 2002a) reports the average number of keystrokes required to enter a character, for example *home* followed by a space on a standard T9 mobile phone requires 6 keystrokes – 4663\* where \* is the next suggestion key, giving a KSPC for *hello* of 5/4=1.25. As with ARP and DA the value is normally averaged over a large corpus of appropriate text for the target language. A KSPC value of 1.0 indicates perfect disambiguation as the user never needs to type any additional letters, while a higher figure reflects the proportional need for the next key in disambiguation (and a lower level, successful word completion). Full-sized non-ambiguous keyboards achieve KSPC=1.00, standard date-stamp method for entering text on 3 keys achieves KSPC=6.45, date-stamp like interaction on 5 keys achieves KSPC=3.13 and multitap on a standard 9-key

mobile phone achieves a KSPC of around 2.03 (MacKenzie, 2002a). Hasselgren et al. (Hasselgren et al., 2003) reported KSPC of 1.01 and 1.08 for T9 using Swedish news and SMS corpora respectively, improving to 0.88 and 1.01 respectively for their bigram model with word completion. KSPC does take into account ranked list position for all words and compares easily with non-predictive text entry approaches; however, it is a rather abstract measure being based on letters, especially for dictionary-based approaches that are inherently word-based.

To gain an insight into potential expert user behaviour with different keyboards, different approaches have been taken to modelling interaction in order to predict expert (trained, error-free) performance. There are two basic approaches: physical movement modelling and keystroke level modelling. We (Dunlop & Crossan, 2000) proposed a keystroke level model based on Card, Moran and Newall's work (S. K. Card, Moran, & Newell, 1980). Our model was based on predicting the time  $T(P)$  taken by an expert user to enter a given phrase. The model calculates this in an equation that combines a set of small time measurements for elements of the user interaction. In the case of text entry: the homing time for the user to settle on the keyboard  $T_h$  (0.40 seconds); the time it takes a user to press a key  $T_k$  (0.28s); the time it takes the user to mentally respond to a system action  $T_m$  (1.35s); the length of an average word  $k_w$  (in our study, 4.98); and the number of words in the phrase  $w$  (in our model, 10). In addition, for predictive text entry where disambiguation occurs by the user moving through the ranked list of suggestions, the ARP value is required here given as  $a=1.03$ ). The overall time equation for entering a phrase is then given as follows:

$$T(P) = T_h + w(k_w T_k + a(T_m + T_k))$$

*Equation 1: Dunlop and Crossan's keystroke model*

This model, as reported in (Dunlop & Crossan, 2000) and corrected by (Pavlovych & Stuerzlinger, 2004), predicts a text entry time for a 10 word phrase at 31.2 seconds, equating to a speed of 19.3wpm – matching closely with experienced user experiments with T9 of 20.4 wpm (James & Reischel, 2001).

In this work we modelled keystroke speed at 0.28s based on a fixed figure from Card et al.'s work that is equivalent to "an average non-secretary typist" on a full QWERTY keypad. This gives fairly accurate predictions but cannot take into account fine grained keyboard design elements that can have a considerable impact on typing speed in practice: for example different keyboard layouts clearly affect the average time it takes a

user to move his/her fingers to the correct keys. Mackenzie's group have conducted considerable work using Fitt's law (Fitts, 1954) to calculate the limit of performance given distance between keys (e.g. (Silfverberg, MacKenzie, & Korhonen, 2000)). The basic form of their distance-based modelling predicts 40.6 wpm for thumb-based predictive input – assuming no *next* key operations (essentially equivalent to no thinking or homing times in equation 1). Later work modifies the Fitt's distance models to take into account two inaccuracies that can noticeably affect predictions: repeated letters on the same key (Soukoreff & MacKenzie, 2002) and parallel finger movements where users move one finger at the same time as pressing with another (MacKenzie & Soukoreff, 2002).

These models are useful in predicting performance but focus on expert error-free performance. More complex modelling approaches have been researched to support novices to model more complete interaction, and to model error behaviour (e.g. (How & Kan, 2005; Pavlovych & Stuerzlinger, 2004; Sandnes, 2005)). Although users studies are the acid test for any interactive system, these models are valuable either in the early stages of design or to understand methods where user experiments are difficult, e.g. by being biased by users' prior experience of current technologies.

### User studies

Models that predict text entry performance only give us part of the picture, proper user studies often give a truer indication of how text entry methods perform in reality. While there are many parameters that can affect the design of user studies, the two prominent issues for text entry experiments are the environment in which the experiments are conducted and the phrases that users enter.

Most user studies into text entry have been conducted in laboratories. A laboratory is a controlled environment that leads to a more consistent user experience than the real world and, thus, considerably easier statistical analysis as there are fewer confounding variables from the environment to interfere with measurements taken. However, conducting experiments on people entering text on mobile phones in quiet office-like settings where they can focus exclusively on the text entry tasks is arguably not representative of normal use! There is a growing debate in mobile HCI research on the validity of laboratory experiments with some researchers arguing that, while the focus of most common errors is different in the real world, laboratory experiments do not miss errors that are found in real-world experiments (Kaikkonen, Kekäläinen, Cankar,

Kallio, & Kankainen, 2005) while others claim a wider range of errors were found in the real-world than in laboratories (Duh, Tan, & Chen, 2006). (Kjeldskov & Graham, 2003) report that "71% [of studied evaluations were] done through laboratory experiments, 19% through field experiments and the remaining 10% through surveys". As a specific example, (Brewster, 2002) showed usability and text entry rates were significantly reduced for users performing an outdoor walking circuit, while entering on a soft numeric key-pad, than those conducting the same experiment in a traditional laboratory. Whereas (Mizobuchi, Chignell, & Newton, 2005) showed that, while walking was slowed down when using a device, it did not impinge upon the text entry rate or accuracy. Some researchers have tried to get the best of both worlds by conducting experiments indoors that attempt to mimic some of the distractions from the real world, while maintaining the controlled laboratory environment (e.g. Lumsden, Kondratova, and Durling 2007).

In experiments users are typically required to enter a set of phrases on devices to measure their text entry speed. While there is no widespread agreement on phrases that are used, (MacKenzie & Soukoreff, 2003) proposed a standard set of short phrases that has been used by other researchers and provides a valuable baseline for comparisons. One problem with mobile phone text entry is that it is often used for short casual messages and testing with formal phrases from a traditional text corpus is not appropriate (see differences discussed above for those who have experimented with formal/newspaper English and SMS). This is compounded by the original multi-tap text entry approach and short length of SMS messages leading to considerable use of, often obscure, abbreviations that are not normally found in a corpus. To address this, (How & Kan, 2005) developed a large set of phrases extracted from SMS users' real text conversations. Although somewhat skewed to local Singaporean phrases and abbreviations (much of *SMS speak* is heavily localised and even personalised within a group of friends), the corpus is a valuable insight into the language often used on mobile phones.

Finally, it should also be noted that entry speeds of 33wpm for users when transcribing text on desktop keyboards have been found to drop to around 19wpm for composing new text (Karat, Halverson, Horn, & Karat, 1999), so most results from text entry experiments can be assumed to be over-inflating speeds by around 40% as they are typically based on transcription.

## CONCLUSIONS AND FUTURE TRENDS

This paper has reviewed a large number of text entry methods that range from standard methods

that are very close to desktop keyboards, through slight variations, to radical novel interaction designs. We have looked at different hardware keyboard designs, different on-screen keyboard layouts, handwriting-based approaches and more novel approaches such as gestures. We have also looked at ambiguous and unambiguous designs, and the related approaches to disambiguation. Much of the work reported has experimental backing to show the potential benefits of each approach. However, when comparing the wide diversity of approaches in the literature to widely available implementations on real devices, the overriding message we see is that guessability, the initial pick-up-and-use usability of hardware/software, is paramount to success.

It is extremely hard to predict future trends for mobile devices: while there is considerable research showing the benefits and strengths of different approaches, market forces and the views of customers and their operators have a major role in deciding which techniques become widely adopted. Predicted gains in expert text entry performance are of no use if people do not understand how to use the text entry approach out of the box. To this end, we see considerable scope for entry methods that provide a smooth transition from novice to expert performance: XT9™ is one successful example of novice-to-expert support, as users get faster they'll learn to be sloppier and type faster, without necessarily being consciously aware of why. Context-aware word completion that learns about individuals is another area that shows good potential: good for slow novice typists as they start, but building context and personalising as they gain proficiency.

Finally, looking at current market directions and the increasing desire to enter more text on small devices, we see the 12-key keypad slowly disappearing from phones to be replaced with less number-centric entry methods. Despite its sub-optimality and problems on small devices, both market trends and some user tests point to the QWERTY keyboard taking on this role, either as a physical or an on-screen keyboard.

## REFERENCES

- Bellman, T., & MacKenzie, I. S. (1998). *A probabilistic character layout strategy for mobile text entry*. Paper presented at the Graphics Interface '98, Toronto, Canada.
- Brewster, S. A. (2002). Overcoming the Lack of Screen Space on Mobile Computers. *Personal and Ubiquitous Computing*, 6(3), 188-205.
- Card, S., Moran, T., & Newell, A. (1983). *The psychology of human-computer interaction*. Lawrence Erlbaum Associates.

- Card, S. K., Moran, T. P., & Newell, A. (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM* 23(7), 396-410.
- Clarkson, E., Clawson, J., Lyons, K., and Starner, T. (2005). An empirical study of typing rates on mini-QWERTY keyboards. In *CHI '05 Extended Abstracts*, ACM Press.
- Cockburn, A., & Siresena, A. (2003). *Evaluating Mobile Text Entry with the Fastap Keypad*. Paper presented at the People and Computers XVII (Volume 2): British Computer Society Conference on Human Computer Interaction, Bath, England.
- Duh, H. B., Tan, G. C., & Chen, V. H. (2006). *Usability evaluation for mobile device: a comparison of laboratory and field tests*. Paper presented at the 8th Conference on Human-Computer interaction with Mobile Devices and Services: MobileHCI '06, Helsinki, Finland.
- Dunlop, M. D. (2004). *Watch-Top Text-Entry: Can Phone-Style Predictive Text-Entry Work With Only 5 Buttons?* Paper presented at the 6th International Conference on Human Computer Interaction with Mobile Devices and Services: MobileHCI 04, Glasgow.
- Dunlop, M. D., & Crossan, A. (2000). Predictive text entry methods for mobile phones. *Personal Technologies*, 4(2).
- Dunlop, M. D., Glen, A., Motaparti, S., & Patel, S. (2006). *Apaptxt: contextually adaptive text entry for mobiles*. Paper presented at the 8th Conference on Human-Computer interaction with Mobile Devices and Services: MobileHCI '06, Helsinki, Finland.
- Engelbart, D. C., & English, W. K. (1968). *A research center for augmenting human intellect*. Paper presented at the AFIPS 1968 Fall Joint Computer Conference, San Francisco, USA.
- Evreinova, T., Evreino, G., & Raisamo, R. (2004). *Four-Key Text Entry for Physically Challenged People*. Paper presented at the 8th ERCIM workshop: User Interfaces for All (Adjunct workshop proceedings), Vienna, Austria.
- Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6), 381-391.
- Goldberg, D., & Richardson, C. (1993). *Touch-typing with a stylus*. Paper presented at the INTERCHI93.
- Gong, J., & Tarasewich, P. (2005). *Alphabetically constrained keypad designs for text entry on mobile devices*. Paper presented at the CHI '05: SIGCHI Conference on Human Factors in Computing Systems Portland, USA.
- Grover, D. L., King, M. T., & Kushler, C. A. (1998). USA Patent No. 5818437. I. Tegic Communications.
- Hasselgren, J., Montnemery, E., Nugues, P., & Svensson, M. (2003). *HMS: A Predictive Text Entry Method Using Bigrams*. Paper presented at the Workshop on Language Modeling for Text Entry Methods, 10th Conference of the European Chapter of the Association of Computational Linguistics, Budapest, Hungary.
- How, Y., & Kan, M.-Y. (2005). *Optimizing predictive text entry for short message service on mobile phones*. Paper presented at the Human Computer Interfaces International (HCII 05), Las Vegas, USA.
- James, C. L., & Reischel, K. M. (2001). *Text input for mobile devices: comparing model prediction to actual performance*. Paper presented at the SIGCHI conference on Human factors in computing systems, Seattle, Washington, United States.
- Kaikkonen, A., Kekäläinen, A., Cankar, M., Kallio, T., & Kankainen, A. (2005). Usability Testing of Mobile Applications: A Comparison between Laboratory and Field Testing. *Journal of Usability Studies*, 1(1), 4-17.
- Karat, C., Halverson, C., Horn, D., & Karat, J. (1999). *Patterns of Entry and Correction in Large Vocabulary Continuous Speech Recognition Systems*. Paper presented at the ACM CHI99: Human Factors in Computing Systems.
- Kjeldskov, J., & Graham, C. (2003). *A Review of Mobile HCI Research methods*. Paper presented at the Fifth International Symposium on Human Computer Interaction with Mobile Devices and Services: MobileHCI 03, Udine, Italy.
- Kushler, C. (1998). *AAC Using a Reduced Keyboard*. Paper presented at the Technology and Persons with Disabilities Conference, USA.
- Lumsden, J., Kondratova, I., Durling, S., Investigating Microphone Efficacy for Facilitation of Mobile Speech-Based Data Entry, *Proceedings of The 21st British Computer Society (BCS) HCI Annual Conference*. Lancaster, UK. 3-7 September 2007.
- MacKenzie, I. S. (2002a). *KSPC (keystrokes per character) as a characteristic of text entry techniques*. Paper presented at the MobileHCI 02: Fourth International Symposium on Human-Computer Interaction with Mobile Devices, Pisa, Italy.
- MacKenzie, I. S. (2002b). *Mobile text entry using three keys*. Paper presented at the Proceedings of the second Nordic conference on Human-computer interaction, Aarhus, Denmark.
- MacKenzie, I. S., Kober, H., Smith, D., Jones, T., & Skepner, E. (2001). *LetterWise: prefix-based disambiguation for mobile text input* Paper presented at the 14th annual ACM symposium on

- User interface software and technology Orlando, USA.
- MacKenzie, I. S., Nonnecke, R. B., McQueen, C., Riddersma, S., & Meltz, M. (1994). *A comparison of three methods of character entry on pen-based computers*. Paper presented at the Factors and Ergonomics Society 38th Annual Meeting, Santa Monica, USA.
- MacKenzie, I. S., & Soukoreff, R. W. (2002). *A model of two-thumb text entry*. Paper presented at the Graphics Interface 2002.
- MacKenzie, I. S., & Soukoreff, R. W. (2003). *Phrase sets for evaluating text entry techniques*. Paper presented at the ACM Conference on Human Factors in Computing Systems – CHI 2003, New York, USA.
- MacKenzie, I. S., Zhang, S. X., & Soukoreff, R. W. (1999). Text entry using soft keyboards. *Behaviour & Information Technology*, 18, 235-244.
- Mankoff, J., & Abowd, G. D. (1998). *Cirrin: a word-level unistroke keyboard for pen input*. Paper presented at the 11th Annual ACM Symposium on User interface Software and Technology: UIST '98., San Francisco, USA.
- Matias, E., MacKenzie, I. S., & Buxton, W. (1996). One-handed touch typing on a QWERTY keyboard. *Human-Computer Interaction*, 11(1), 1-27.
- Mizobuchi, S., Chignell, M., & Newton, D. (2005). *Mobile text entry: relationship between walking speed and text input task difficulty*. Paper presented at the MobileHCI 05: 7th international Conference on Human Computer interaction with Mobile Devices and Services Salzburg, Austria.
- Norman, D. A. (2002). *The Design of Everyday Things: Basic Books*.
- Norman, D. A., & Fisher, D. (1982). Why Alphabetic Keyboards Are Not Easy to Use: Keyboard Layout Doesn't Much Matter. *The Journal of the Human Factors Society*, 25(5), 509-519.
- Pavlovych, A., & Stuerzlinger, W. (2004). *Model for non-expert text entry speed on 12-button phone keypads*. Paper presented at the CHI '04: SIGCHI Conference on Human Factors in Computing Systems, Vienna, Austria.
- Perlin, K. (1998). *Quikwriting: continuous stylus-based text entry*. Paper presented at the 11th Annual ACM Symposium on User interface Software and Technology: UIST '98, San Francisco, USA.
- Proschowsky, M., Schultz, N., & Jacobsen, N. E. (2006). *An intuitive text input method for touch wheels*. Paper presented at the SIGCHI Conference on Human Factors in Computing Systems: CHI '06, Montréal, Canada.
- Robinson, B. A., & Longe, M. R. (2000). USA Patent No. 6,801,190 / 7,088,345.
- Sandnes, F. E. (2005). *Evaluating mobile text entry strategies with finite state automata*. Paper presented at the MobileHCI 05: 7th international Conference on Human Computer interaction with Mobile Devices and Services, Salzburg, Austria.
- Shannon, C. (1951). Prediction and Entropy of Printed English. *Bell System Technical Journal*, 30, 50-64.
- Silfverberg, M., MacKenzie, I. S., & Korhonen, P. (2000). *Predicting text entry speed on mobile phones*. Paper presented at the SIGCHI Conference on Human Factors in Computing Systems (CHI'00), The Hague, The Netherlands.
- Soukoreff, R. W., & MacKenzie, I. S. (2002). *Using Fitts' law to model key repeat time in text entry models*. Paper presented at the Graphics Interface 2002.
- Ward, D. J., Blackwell, A. F., & MacKay, D. J. (2000). *Dasher - a data entry interface using continuous gestures and language models*. Paper presented at the 13th Annual ACM Symposium on User interface Software and Technology: UIST '00, San Diego, USA.
- Wigdor, D., & Balakrishnan, R. (2003). *TiltText: using tilt for text input to mobile phones*. Paper presented at the 16th annual ACM Symposium on User Interface Software and Technology (UIST '03), New York, USA.
- Wigdor, D., & Balakrishnan, R. A. (2004). *Comparison of consecutive and concurrent input text entry techniques for mobile phones*. Paper presented at the SIGCHI Conference on Human Factors in Computing Systems (CHI '04), New York, USA.
- Williamson, J., & Murray-Smith, R. (2005). *Dynamics and probabilistic text entry*. Paper presented at the Hamilton Summer School on Switching and Learning in Feedback systems.
- Zhai, S., & Kristensson, P. (2003). *Shorthand writing on stylus keyboard*. Paper presented at the SIGCHI Conference on Human Factors in Computing Systems: CHI '03, Ft. Lauderdale, Florida, USA.

## ACKNOWLEDGEMENTS

Our thanks are extended to our reviewers whose comments have strengthened this paper. We are also grateful to those organisations and companies that have given permission to use images throughout the paper.

All trademarks are used with appropriate reference to their holders. FrogPad is a trademark of FrogPad Inc. – Figure 2 (right) is used with permission of FrogPad. FastTap is a trademark of Digit Wireless Inc. Figure 3 used with permission of Digit Wireless. Douglas Engelbart image

(Figure 4 left) is used courtesy of SRI International, Menlo Park, CA. MicroWriter Agenda image (Figure 4 right) used with permission of Bellaire Electronics. Blackberry is a trademark of Research In Motion Limited. Fitaly keyboard image (Fig 6 left) used with permission of Textware Solutions. XT9 is a registered trademark of Tegic Communications, Inc. The Tegic Communications, Inc. name, icons, trademarks and image in Figure 7 are used with permission. Microsoft, Windows is a registered trademark of Microsoft Corp. Unistroke is patented technology

of Xerox. Graffiti is a registered trademark of Palm Inc; Jot and WordComplete of CIC. Figure 9 (centre and right) images used courtesy of KeyPoint Technologies of whom Apaptxt is a trademark. Figure 10, 11 and 12 images used courtesy of related paper authors. Figures 1, 2 (left), 6 (right) and 8 drawn by authors and may be altered from original layouts for clarity. Figures 9 (left), 10 (left) and 12 screen images taken by authors. Images in Figure 5, original photographs by authors.