From Wax Tablets to Touchscreens: An introduction to text-entry research

How we can enable users to transmit text to mobile and ubiquitous computer systems as quickly and as accurately as possible.

By Per Ola Kristensson
DOI: 10.1145/2659835

Text entry is an integral activity in our society. Text-entry methods not only allow us to communicate asynchronously in a variety of media—such as email, books, social media, and electronic documents—but contribute to an accurate recording of our culture and history for future generations. In addition, text-entry methods in the form of augmentative and alternative communication (AAC) devices allow nonspeaking individuals with motor disabilities to communicate in real-time via speech synthesis.

Writing methods have been researched for thousands of years. One famous example is the shorthand system *notæ Tironianæ* ("the Tironian notes"), invented by Marcus Tullius Tiro in the 60s B.C. Tiro's shorthand system was originally used by secretaries in Rome and was gradually expanded to about 13,000 shorthand symbols. Writers carved symbols into wax-covered tablets using a stylus [1]. Another example is *nova ars notaria* ("the new note art"), believed to have been invented in England in the 13th century. This shorthand system is interesting because we have some insights into its design process. The system encoded letters in the alphabet as simple line marks and common word stems as line marks for the initial letter, followed by a sequence of dots and lines that served to differentiate different word stems sharing the same initial letter. The common word stems were identified via frequency analysis in the Book of Psalms. In other words, two key principles of text entry design have been known for more than 800 years: (1) minimize the time it takes for users to articulate their intended text, and (2) exploit statistical regularities in natural languages via language modeling [2].

Typewriters eventually replaced shorthand systems. Following a period of intense competition, typewriter design converged on the now ubiquitous QWERTY layout, invented by Christopher Sholes in 1868 and today transplanted into the vast majority of the world's desktop, laptop, and touchscreen keyboards [3, 4]. In the last decades, several new text-entry methods have been designed for a variety of mobile and ubiquitous devices [5]. Additionally, researchers have been exploring new efficient AAC strategies for nonspeaking individuals with motor disabilities [6].

However, text entry remains suboptimal in many situations, in particular when users are situation-impaired, for example, because they are encumbered or mobile, or rate-limited by,

Image by Sergey Nivens


-VIDEO
-MUSIC
-FILMS
-SEARCH
-CONTACTS
-MESSAGES
TUTORIALS
INVESTMENT
NETWORKING
STRATEGY
PLAN

for instance, a motor disability. Fortunately, recently developed sensors and data processing approaches have enabled text-entry researchers to reimagine the way we write, promising a more fluid and efficient writing process for a variety of situations. This article provides an introduction to text-entry research, with a particular focus on emerging intelligent text-entry methods. Such techniques use methods from artificial intelligence and machine learning to infer or predict what a user is intending to write; see Karat et al. for an overview [7].

INTELLIGENT TEXT-ENTRY METHODS

A well-defined intelligent text-entry method can typically be modeled probabilistically. Assume the user provides us with an observation sequence articulating the user's intended text. Examples of such an observation sequence include a sequence of pressed keys on a keyboard, a series of touch points on a capacitive touchscreen, an audio recording of the user's speech, etc. Given an observation sequence O, the objective is to identify the most likely word sequence $\widehat{W}$ that maximizes the posterior probability $P(W|O)$ under the model:

$$\widehat{W} = \arg\max_{W} P(W|O) = \arg\max_{W} \frac{P(O|W)P(W)}{P(O)}$$

The denominator $P(O)$ is invariant in a search for the highest probability word sequence and can therefore be ignored. $P(O|W)$ is the likelihood of the observation sequence given a word sequence and $P(W)$ is the prior probability of a word sequence, which is computed using a language model. A language model assigns probabilities to word sequences.

In other words, an intelligent text-entry method identifies the most likely word sequence by combining the probabilities of two models: a likelihood model of the observation sequence and a language model. The objective of designers of intelligent text-entry methods is to create suitable likelihood and language models that maximize the probability that the text-entry method can infer or predict the user's intended word sequence.

An example of an intelligent textentry method is our work on what is now academically known as the "gesture keyboard" [8, 9]. A gesture keyboard encodes words as geometric word trajectories over a keyboard layout. For instance, the word trajectory for the word "the" starts at the center of the T key and extends to the centers of the *H* and *E* keys in sequence. Double letters are ignored, which means words such as "the" and "thee" are ambiguous. The intended word is disambiguated with the aid of a language model.

As illustrated in Figure 1, the geometric word trajectories defined by a gesture keyboard become an encoding system for words. A user writes words by articulating gestures that resemble the word trajectories. For instance, in Figure 2, the user has produced a gesture for the word "rabbit." A gesture recognizer then computes the likelihood of a user's gesture being similar to a word trajectory. This is the likelihood model of the gesture keyboard. The likelihood of a word given the observation sequence is then combined with the prior probability of a word. The prior probability is computed with the aid of a statistical language model, which assigns probabilities to word sequences; see Kristensson and Zhai [8], for a detailed technical description.

Gesture keyboards provide a seamless transition from novice to expert behavior. A novice user can write words by tracing out the words on the touchscreen keyboard. Behaviorally, this is very similar to touchscreen keyboard typing. However, as the user writes words with a gesture keyboard, the motor patterns for individual words gradually consolidate in the user's motor memory. After a sufficient amount of repetition, the user can quickly articulate the gesture for a word by direct recall from motor memory. At this point, a user can write words faster using a gesture keyboard compared to a regular touchscreen keyboard, as direct recall from motor memory does not demand visual attention [8, 9].

Our work on gesture keyboards began in 2001. The first public gesture keyboard was released on IBM alphaWorks in 2004. At this point in time, it was called IBM SHARK Shorthand. In

Figure 1. Word trajectories for the words "and," "the," and "this" traced on a QWERTY layout.

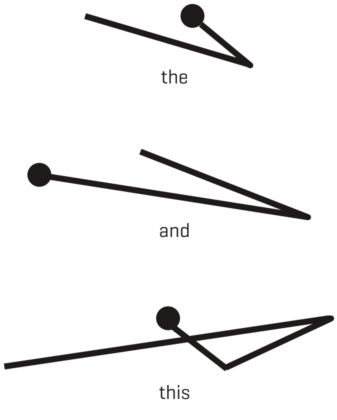
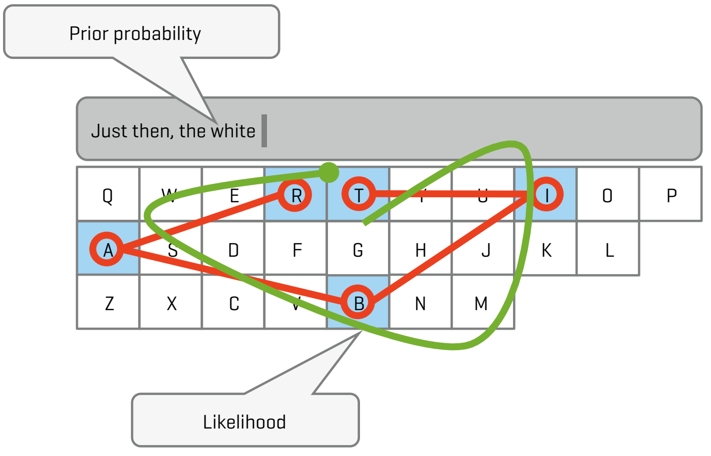


Figure 2. A user has articulated the gesture for the word "rabbit" in a gesture keyboard interface.



2007, we created a tech startup called ShapeWriter, Inc. and released the first gesture keyboard for iPhones in 2007 and for Google Android in 2008. In 2010, our company was acquired by Nuance Communications. Today, gesture keyboard technology is available by default on all Google Android devices, where it is known as "continuous input" or "gesture typing."

Besides the gesture keyboard, other examples of intelligent text-entry methods include speech and handwriting recognition systems, keyboard auto-correction algorithms, and predictive AAC devices.

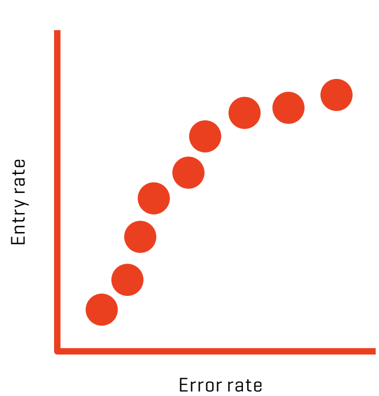
Image by Andrei Rancz

DESIGN PRINCIPLES

The text-entry design space is highly multidimensional and may include design constraints as diverse as device form factor, ergonomics, production cost, whether a user must be able to use it with one hand, etc. Nevertheless, regardless of individual design constraints, most text-entry methods are designed to satisfy two primary objectives: (1) a high effective entry rate, and (2) a fast learning curve.

Effective Entry Rate. One of the most critical design objectives of a text-entry method is to provide users with a high-entry rate while simultaneously ensuring the error rate is below some tolerance threshold. Entry rate is measured in words-per-minute (wpm), with a word defined as five consecutive characters including space. Error rate can be measured in various ways. One well-defined measure is character error rate (CER), which is the minimum edit distance between the intended text and the actual written text, divided by the number of characters in the intended text.

Figure 3. An illustration of a text entry performance envelope.



Entry and error rates are measured in controlled experiments. The researchers recruit representative participants from the population and instruct them to write prompted memorable stimulus sentences as quickly and as accurately as possible [10, 11]. This task is known as the "transcription task." Since text entry performance is a complex skill, it is advisable to test a participant's text entry performance over multiple sessions spread out over multiple days. Finally, while the transcription task is the de-facto standard experimental task in text entry, we have recently demonstrated it is also possible to use a "composition task," in which users are asked to compose brief messages [12]. This latter task might be of interest when it is desirable to test text entry methods in a more realistic setting outside a lab environment.

The entry and error rate of a text entry method are coupled via a speed-accuracy trade-off. A high entry rate is meaningless if the error rate is very high. Conversely, an accurate but very slow text-entry method is of limited use to most people. Since entry and error rates are interrelated, it is not meaningful to analyze them in isolation. Figure 3 visualizes the performance envelope of a hypothetical text entry method by plotting entry rate as a function of error rate.

To easily compare text-entry methods, it is often meaningful to define the effective entry rate, which is the writing speed of a text-entry method when error rate is controlled within a certain tolerance threshold, such as a 2 percent CER. The effective entry rate is typically measured empirically by sampling text-entry performance from a representative set of participants. Sometimes it is also possible to use mathematical modeling to predict text entry performance [13].

While it is clear that a high effective entry rate is desirable, it is less clear how fast a text-entry method needs to be for it to be able to fully support users. We have recently proposed to approach this problem via a new construct we call the "inviscid entry rate" [14]. A text-entry method is said to be inviscid when the user's creativity is the bottleneck rather

than the text entry method. At this point, the text-entry method is rate-limited, the time it takes the user to conceive the text and further reductions in the time it takes to articulate text are bound to provide only negligible speed improvements. For mobile text entry, we have empirically estimated the sufficiently inviscid entry rate to be around 67 wpm. Currently no known mobile text-entry method provides such a high entry rate, which means mobile text-entry methods still have room to improve [14].

In order to increase the effective entry rate, the text-entry method needs to reduce the time it takes for users to articulate their intended text. One straightforward idea is to provide users with word predictions. Another, perhaps less obvious, example of how this principle can be adhered to is the work on optimized stylus keyboards for mobile devices [15, 16]. When a user operates a stylus keyboard, the user holds the mobile device in one hand and the stylus in the other. The problem with a stylus keyboard based on the traditional QWERTY layout is that a QWERTY layout tends to distribute frequent letter key-pairs to the left and right hand sides, respectively. This means the stylus tends to zigzag between the left and right hand sides of the keyboard, which is clearly suboptimal. An optimized stylus keyboard rectifies this by rearranging the keyboard layout in such a way that the average movement time when the stylus travels between letter keys is minimized.

Another approach to increase the effective entry rate is to enable users to be less precise in their movements when they articulate their intended text. For example, a touchscreen keyboard can use an auto-correction algorithm to allow users to type faster and less precisely [17, 18]

Examples of intelligent text-entry methods include speech and handwriting recognition systems, keyboard auto-correction algorithms, and predictive AAC devices.

In practice errors are unavoidable in text-entry, and error-correction activities tend to substantially lower the entry rate. Therefore, it is also critical to design efficient error-correction methods, in particular for text-entry modalities that have been shown to exhibit severe performance degradations when users attempt to correct errors, such as speech recognition [8].

Familiarity and Path Dependency. Another design consideration is the learning curve of the text-entry method. Text entry is a complex skill that demands extensive practice in order for users to reach their full performance potential. However, users tend to be reluctant to adopt a text-entry method if it requires substantial learning investment. Therefore, users' familiarity with a new text-entry method has been a strong predictor of the success of the text-entry method. This phenomenon is known as path dependency in economics. One of the primary examples of path dependency is the QWERTY keyboard, which, despite being suboptimal for 10-finger touch-typing, has still been "good enough" for users to be reluctant to change to faster competing alternatives; see David [3], and also Liebowitz [19], for a critique .

Optimized keyboards, which rearrange the keys on the keyboard layout, require a substantial training investment by users before they will see any performance benefit. The point when a user becomes able to write faster with a new text-entry method is known as the "crossover point." The more training investment required in order for users to reach the crossover point, the more challenging it will be to persuade users to adopt to the new text-entry method. Although optimized keyboards have been extensively researched in the last 20 years, they have never been widely adopted by users.

RESEARCH QUESTIONS AND FUTURE DIRECTIONS

Text-entry research tends to focus on three primary research objectives:

▶ Designing better text entry methods.

▶ Investigating behavioral aspects of existing text-entry methods.

▶ Contributing to text-entry design and evaluation methodology.

Two emerging themes in text entry research that are likely to attract considerable attention in the near future are flexibility and multimodality and context awareness.

Flexibility and Multimodality. As mobile devices acquire additional sensors and gain computational power, it becomes feasible to allow users more flexibility in text entry. For instance, we have designed a probabilistic algorithm that can flexibly fuse gesture keyboard and speech recognition input [20]. Using this algorithm it is possible, for instance, to fix an erroneous word from the speech-recognition modality by writing the intended replacement word using the gesture keyboard modality. The system automatically locates the erroneous word in the speech-recognition modality and replaces it with the intended word without the need for the user to explicitly indicate the error location. The algorithm achieves this by searching for the most likely word sequence hypothesis through both modalities' statistical hypothesis spaces simultaneously. In a controlled experiment, this error correction style could reduce the relative word error rate by 44 percent [20]. Such flexibility can potentially be extended to different modalities. There is also considerable work to be carried out in order to understand the flexible multimodal text-entry design space and to improve the algorithms that fuse the different hypothesis spaces.

Context Awareness. Another emerging research area in text entry is in designing systems that can leverage contextual information. More advanced mobile device sensing techniques enable text-entry methods to potentially understand much more about the user's overall situation, including the user's location, affective state, and social setting. This information can potentially be used to provide better word predictions, in particular for AAC devices.

Text-entry methods not only allow us to communicate asynchronously in a variety of media, ... but contribute to an accurate recording of our culture and history for future generations.

CONCLUSION

Text entry is a diverse area of research that spans multiple disciplines, including human-computer interaction, natural language processing, augmentative and alternative communication, and signal processing. While the text-entry space is highly multidimensional, the design of a state-of-the-art text-entry method is essentially an optimization problem in which the goal is to simultaneously achieve a high effective entry rate and a fast learning curve. The next-generation text-entry methods use recently developed sensors and data processing approaches in order to create efficient statistical models, amplifying the user's ability to transmit their intended text to computer systems as quickly and as accurately as possible.

References

[1] Melin, O.W. Stenografiens historia, första delen. P.A. Norstedt & Söner, Stockholm, 1927.

[2] Kristensson, P.O. Five Challenges for Intelligent Text Entry Methods. *AI Magazine* 30, 4 (2009), 85-94.

[3] David, P. 1985. Clio and the Economics of QWERTY. *American Economic Review* 75, 2 (1985), 332-337.

[4] Gould, S. J. The Panda's Thumb of Technology. In *Bully for Brontosaurus*. Penguin Books, London, 1992, 59-75.

[5] MacKenzie, I. S., and Soukoreff, R. W. Text Entry for Mobile Computing: Models and methods, theory and practice. *Human-Computer Interaction* 17, 2 (2002), 147-198.

[6] Glennen, S. and DeCoste, D. *The Handbook of Augmentative and Alternative Communication*. Singular Publishing Group, San Diego, 1997.

[7] Karat, C.-M., Halverson, C., Horn, D., and Karat, J. Patterns of Entry and Correction in Large Vocabulary Continuous Speech Recognition Systems. In *Proceedings of the 17th ACM Conference on Human Factors in Computing Systems (CHI 1999)*. ACM Press, New York, 1999, 568-575.

[8] Kristensson, P.O. and Zhai, S. SHARK2: A large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST 2004)*. ACM Press, New York, 2004, 43-52.

[9] Zhai, S. and Kristensson, P.O. The Word-gesture Keyboard: Reimagining keyboard interaction. *Communications of the ACM* 55, 9 (2012), 91-101.

[10] MacKenzie, I.S. and Soukoreff, R.W. Phrase Sets for Evaluating Text Entry Techniques. In *Extended Abstracts of the 21st ACM Conference on Human Factors in Computing Systems (CHI 2003)*. ACM Press, New York, 2003, 754-755.

[11] Vertanen, K. and Kristensson, P.O. A Versatile Dataset for Text Entry Evaluations Based on Genuine Mobile Emails. In *Proceedings of the 13th ACM International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI 2011)*. ACM Press, New York, 2011, 295-298.

[12] Vertanen, K. and Kristensson, P.O. Complementing Text Entry Evaluations with a Composition Task. *ACM Transactions on Computer-Human Interaction* 21, 2 (2014).

[13] Clarkson, E., Lyons, K., Clawson, J., and Starner, T. Revisiting and Validating a Model of Two-thumb Text Entry. In *Proceedings of the 25th ACM Conference on Human Factors in Computing Systems (CHI 2007)*. ACM Press, New York, 2007, 163-166.

[14] Kristensson, P.O. and Vertanen, K. The Inviscid Text Entry Rate and its Application as a Grand Goal for Mobile Text Entry. In *Proceedings of the 16th ACM International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI 2014)*. ACM Press, New York, 2014 (in press).

[15] MacKenzie, I.S. and Zhang, S.X. The Design And Evaluation of a High- Performance Soft Keyboard. In *Proceedings of the 17th ACM Conference on Human Factors in Computing Systems (CHI 1999)*. ACM Press, New York, 1999, 25-31.

[16] Zhai, S., Hunter, M., and Smith, B.A. The Metropolis Keyboard – an exploration of quantitative techniques for virtual keyboard design. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST 2000)*. ACM Press, New York, 2000, 119-128.

[17] Goel, M., Findlater, L., and Wobbrock, J. WalkType: Using accelerometer data to accommodate situational impairments in mobile touch screen text entry. In *Proceedings of the 30th ACM Conference on Human Factors in Computing Systems (CHI 2012)*. ACM Press, New York, 2012, 2687-2696.

[18] Goodman, J., Venolia, G., Steury, K., and Parker, C. Language Models for Soft Keyboards. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI 2002)*. AAAI Press, 2002, 419-424.

[19] Liebowitz, S.J. and Margolis, S. E. The Fable of the Keys. *Journal of Law & Economics* 33, 1 (1990), 1-25.

[20] Kristensson, P.O. and Vertanen, K. Asynchronous Multimodal Text Entry Using Speech and Gesture Keyboards. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association (Interspeech 2011)*. ISCA, 2011, 581-584.

Biography

Per Ola Kristensson is a lecturer in human computer interaction at the University of St Andrews, UK. In 2013, he was recognized as an Innovator Under 35 (TR35) by *MIT Technology Review*. In 2014, he won the Royal Society of Edinburgh Early Career Prize in Physical Sciences, the Sir Thomas Makdougall Brisbane Medal.

© 2014 ACM 1528-4972/14/09 $15.00