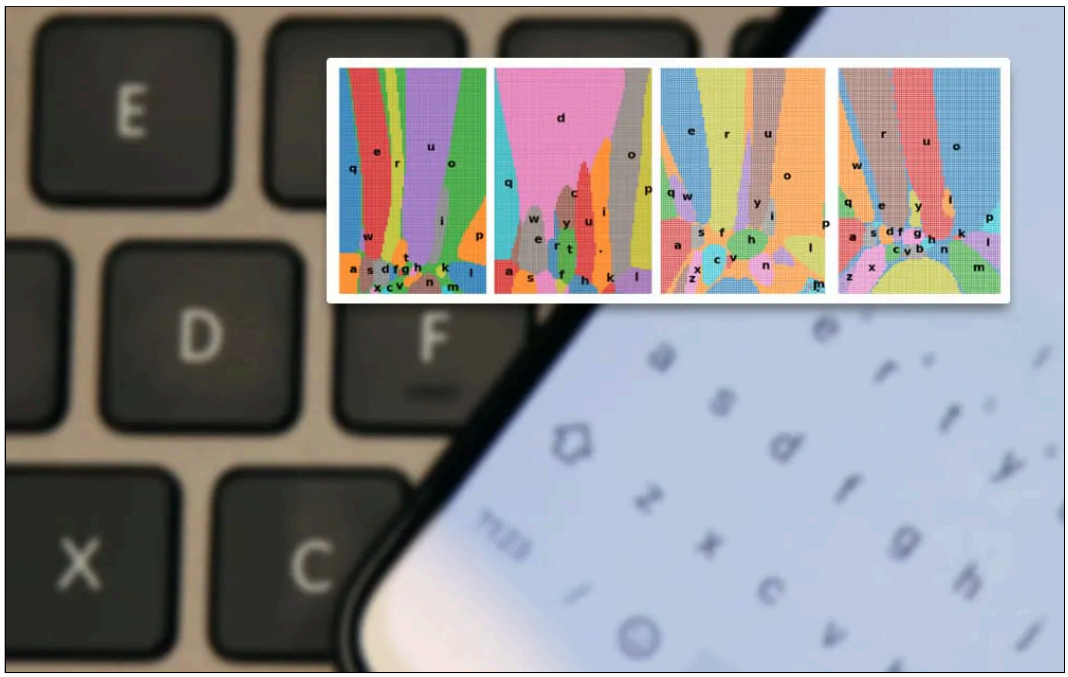
An AI-Driven Invisible Mobile Keyboard That Lets You Type 157% Faster

Originally published at https://www.unite.ai/an-ai-driven-invisible-mobile-keyboard-that-lets-you-type-157-faster/



Researchers from South Korea have used machine learning techniques to develop an 'invisible' keyboard for space-constrained mobile devices that allows users to type 157.5% faster, even though no keyboard is apparent on the screen.

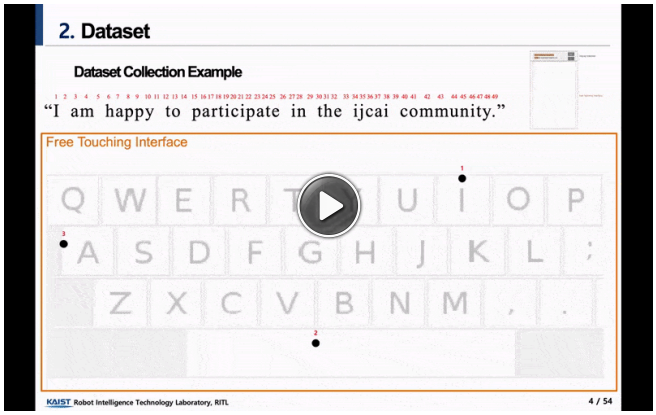
User response to the new method – called simply Invisible Mobile Keyboard (IMK) – is reported to be very positive, with test users reporting low levels of physical, mental and temporal demand while using the keyboard. In terms of efficiency, IMK lightly outstrips the most recent state of the art alternative input method, rising to a vanguard score of 51.6 words per minute.

The Phantom Keyboard

To start generating input, users can simply begin typing on the screen, as if a keyboard was visible (though none is). Nothing pops up to obstruct the view of the content, and the typed words will appear in any receptive text box where the typing originates, and optionally as a thin stream of text that the user can check for accuracy.

The system self-calibrates from the moment it recognizes input. Therefore the user can have the mobile device in landscape or portrait mode, and use the entirety of the available screen space to type out their text.

In an accompanying video (see end of article, and image directly below) the authors of the paper illustrate how the action works, though they clarify that no actual keyboard appears during input (it's only there for illustrative purposes in the video):



CLICK TO VIEW ANIMATED GIF

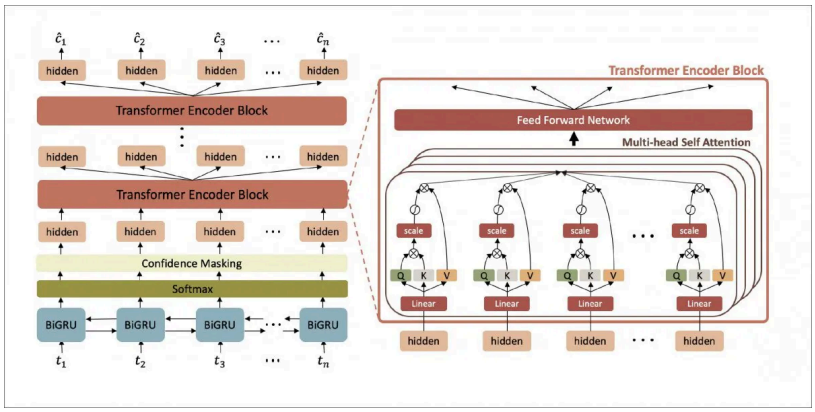
This is an example of IMK at the data collection stage, though it operates identically in end-usage. The keyboard that appears is only for illustrative purposes, and does not appear to the user either during the data collection process or in final usage of the interface. Source: https://www.youtube.com/watch?v=PuhiVGOfIR0

Typing as a Coordinate System

The research originates from the Korea Advanced Institute of Science and Technology (KAIST), and exploits our natural ability to 'plot' where the next key is on a keyboard. Though it may seem counter-intuitive to hide the keyboard and expect a user's finger to find the next desired key, in fact even an average typist instinctively heads for the correct character.

Effectively IMK treats the keyboard as a plot matrix, and the authors have compiled an extensive database of user input in order to supply data for the system's Self-Attention Neural Character Decoder (SA-NCD) to train against.

SA-NCD will note the position of a 'key-fall' and calculate the probability of which key was desired. As words build up through key-strokes, SA-NCD can compile and break up the characters into their constituent intended words, cleaning the input on a live basis.



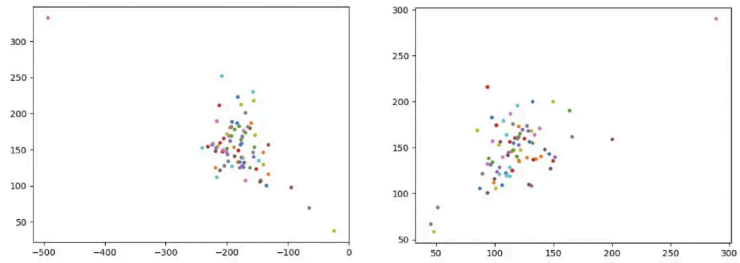
The network architecture of SA-NCD, where Q/K/V stand for query, key and value of self-attention. Source: https://arxiv.org/pdf/2108.09030.pdf

SA-NCD does not wait for completion of a possible sentence, since it has no idea when the sentence input will end, and as a word or words are added to the phrase, it may re-visit and re-write earlier interpretations from the sentence in the light of the latest input.

Database

In order to fuel the training process, the researchers gathered around two million pairs of touch points and text from test subjects, who were using a simple web-based interface accessed from touch-capable mobile devices.

The dataset contains the user's name initials, the screen size of their device, their age, the type of mobile device used (i.e. tablet, smartphone, etc.), and the x and y coordinate values of each registered keyfall.



Average positions of keyfalls among users, with dots of identical color signifying keyfalls from the same users. Identifying same-user data helps to optimize the dat comparing average keyfall groupings from individual users, rather than training one user's keystrokes against each other.

The training had to account for the notable variations in average pixel distance between strokes among users. Some users, perhaps those accustomed to very cramped software keyboards, maintained an average distance between keys of only 50 pixels on the z axis, while others averaged 300 pixels.

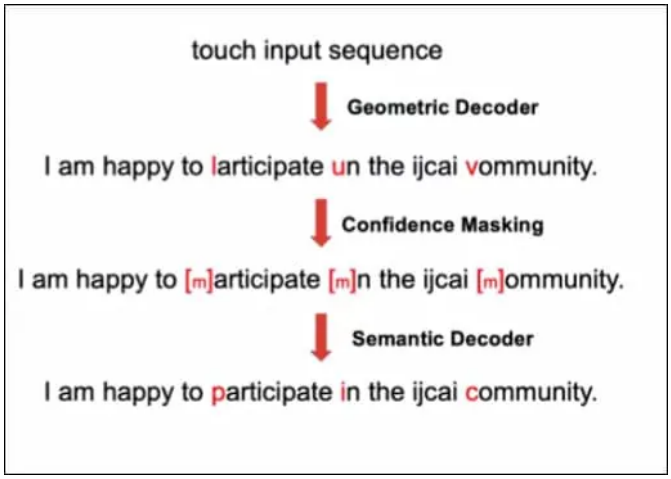
These differences are critical, since in the case of Y axis, an error would place the keyfall on the wrong row, substituting, for instance, an 'I' or an 'M' for the intended 'K' stroke.

Architecture and Training

SA-NCD consists of two decoder modules: a geometric decoder, which calculates where on the invisible keyboard a keystroke was intended to fall; and a semantic decoder, which handles live interpretation of the input text.

The geometric decoder uses Bidirectional GRU (BiGRU), with GRU adopted as a Recurrent Neural Network (RNN), with forward and backward passes facilitating a constantly-changing interpretation of the sentence.

The semantic component uses a Transformer architecture, which interprets input after it has passed through a 'confidence masking' process designed to compare average usage against the new specific keyfall. The semantic decoder was trained as a masked character language model against the One Billion Word Benchmark, a 2014 collaboration between Google, Cambridge University and the University of Edinburgh.

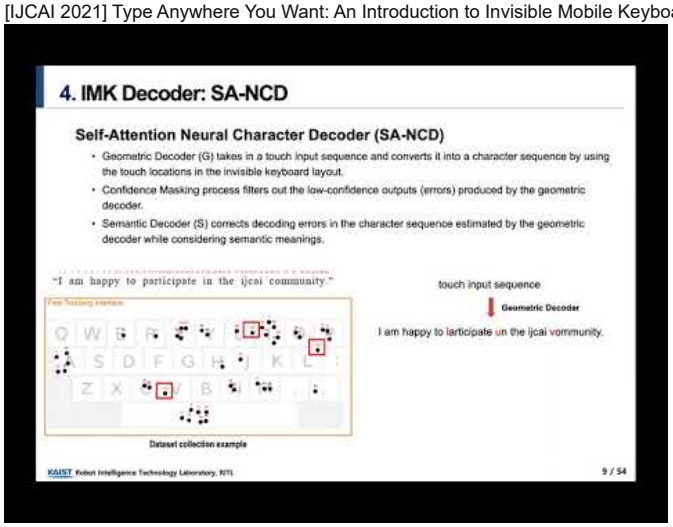


Results

In tests, users were able to type 157.5% faster using IMK than with third-party software keyboards on their own smartphones. Further, it was found that IMK surpassed results obtained by rival novel methods, such as gesture-based, touch-based and ten-finger text entry methods of recent years. The paper reports that users showed high satisfaction with the system.

See the authors' video below in order to find out more about IMK.

[IJCAI 2021] Type Anywhere You Want: An Introduction to Invisible Mobile Keyboard (explained)



Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.
Portfolio site: martinanderson.ai
Contact: martin@martinanderson.ai

Discover More