Lecture 8 Text Entry on Mobile Devices -2

Xiaojun Bi

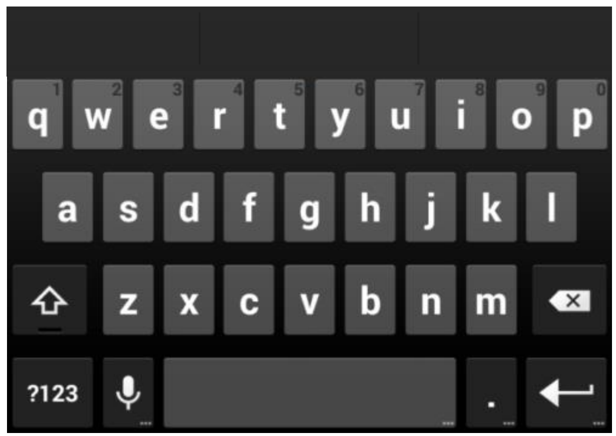
Stony Brook University

xiaojun@cs.stonybrook.edu

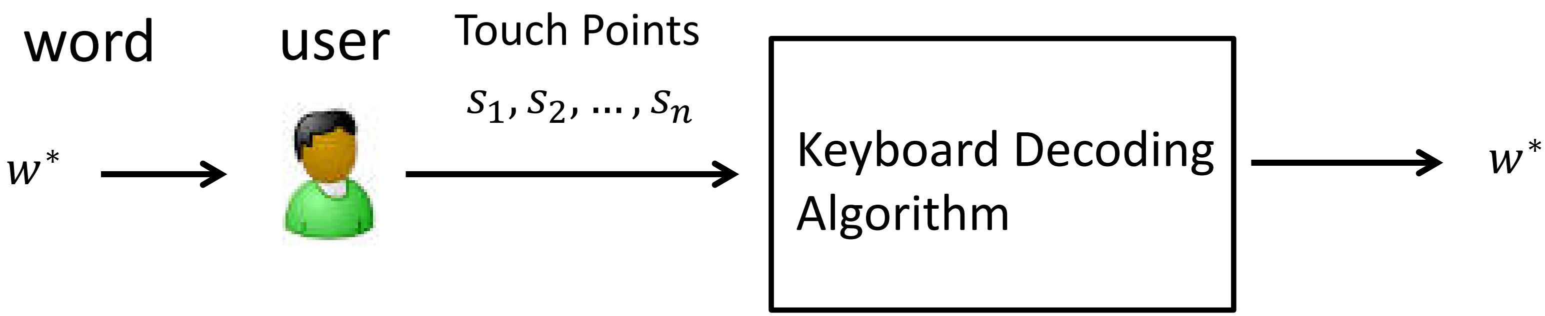
10/3/2019

Smart Touch Keyboard

Typed Word		Keyboard Output
agsim	→	again
quivj	→	quick
fav	→	favorite



Text Entry Decoding Algorithm



$$W^* = \underset{W \in L}{\arg\max} P(W|S).$$

From Bayes' rule: $$P(W|S) = \frac{P(S|W)P(W)}{P(S)}.$$

As P(S) is a constant across all the words, we have:

$$W^* = \underset{W \in L}{\arg\max} P(S|W)P(W).$$

- Spatial Model:

Assuming that W is comprised of n letters: $c_1, c_2, c_3, \ldots, c_n$, S has n touch points, and each tap is independent, we have:

$$P(S|W) = \prod_{i=1}^{n} P(s_i|c_i).$$

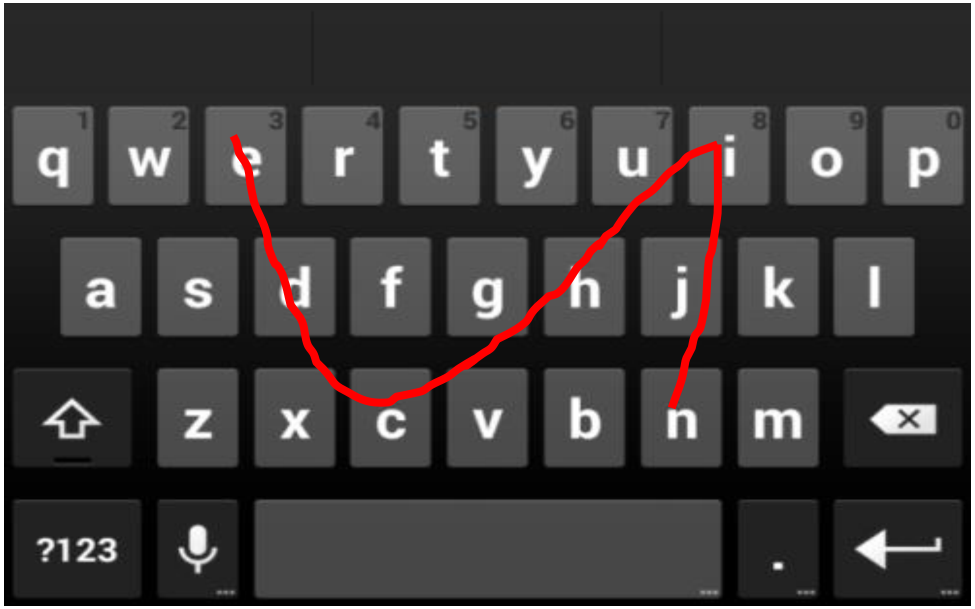
$$P(s_i|c_i) = \frac{1}{2\pi\sigma_{i_x}\sigma_{i_y}\sqrt{1-\rho_i^2}} \exp\left[-\frac{z}{2(1-\rho_i^2)}\right]$$ Gaussian Distribution

- Language Model:

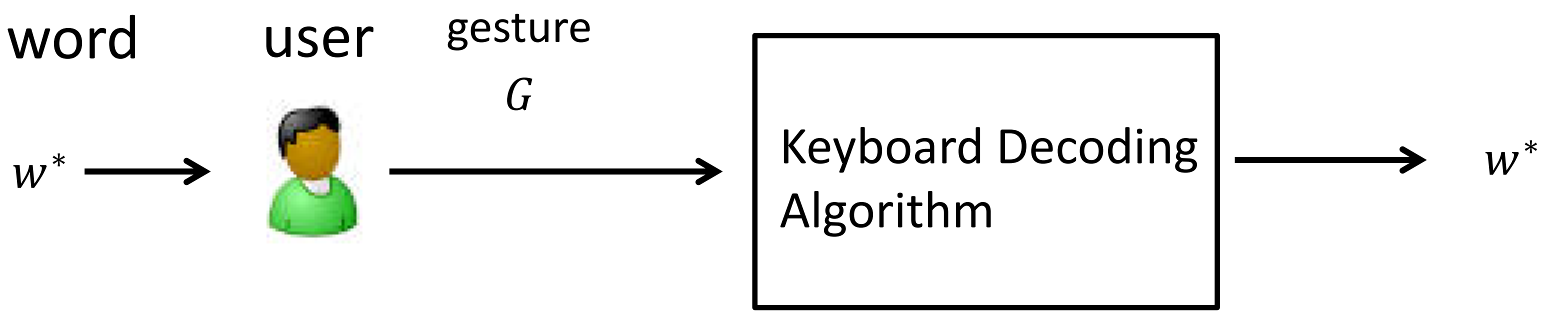
$P(W)$ is obtained from a Language Model (LM)

Gesture Keyboard

Entering *nice*



Gesture Decoder



$$W^* = \underset{w}{\operatorname{argmax}}\, P(W|G) = \underset{w}{\operatorname{argmax}} \frac{P(G|W)P(W)}{P(G)}$$

$$W^* = \underset{w}{\operatorname{argmax}}\, P(G|W)P(W)$$

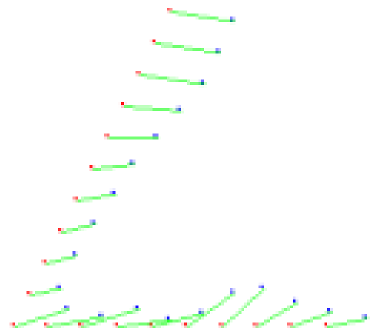
How to calculate $P(G|W)$?

Shumin Zhai and Per Ola Kristensson. 2012. The word-gesture keyboard: reimagining keyboard interaction. Commun. ACM 55, 9 (September 2012),

SHARK^2 Algorithm

- Location Recognition Channel

$$x_s = \frac{1}{N}\sum_{i=1}^{N}\left\| u_i - t_i \right\|_2$$



- Shape Matching Channel

Per-Ola Kristensson and Shumin Zhai. 2004. SHARK^2: a large vocabulary shorthand writing system for pen-based computers. In *UIST '04*. 43-52.

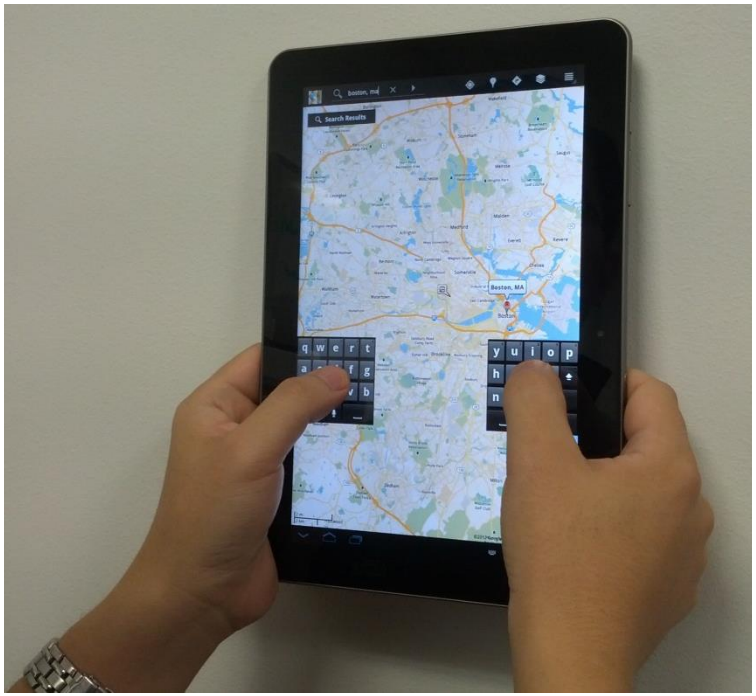
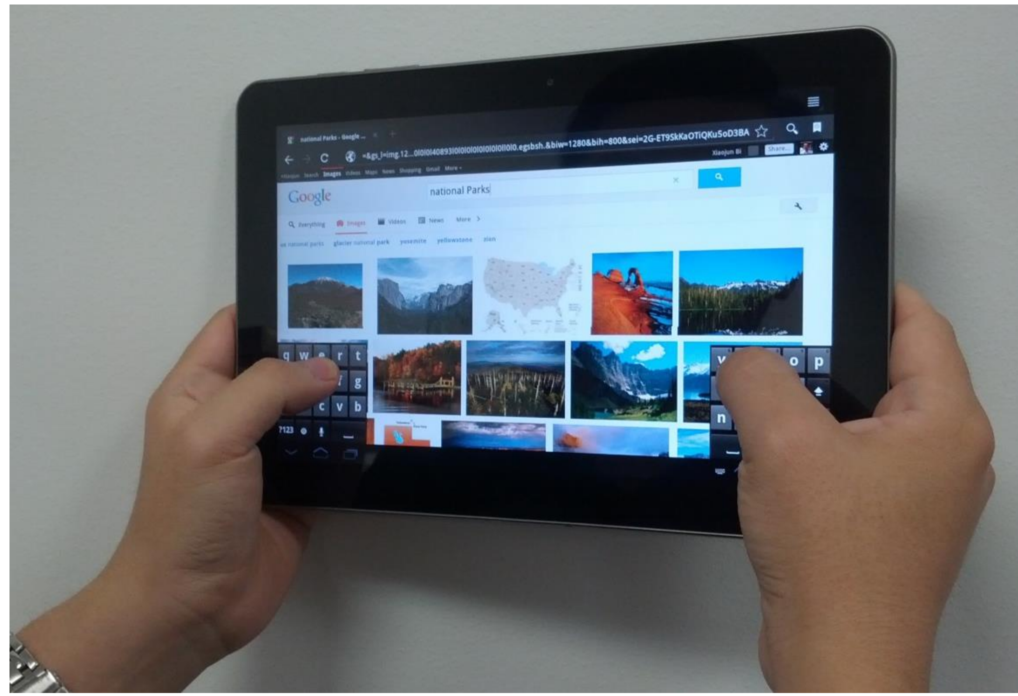
Gesture Keyboard

ShapeWriter

Android

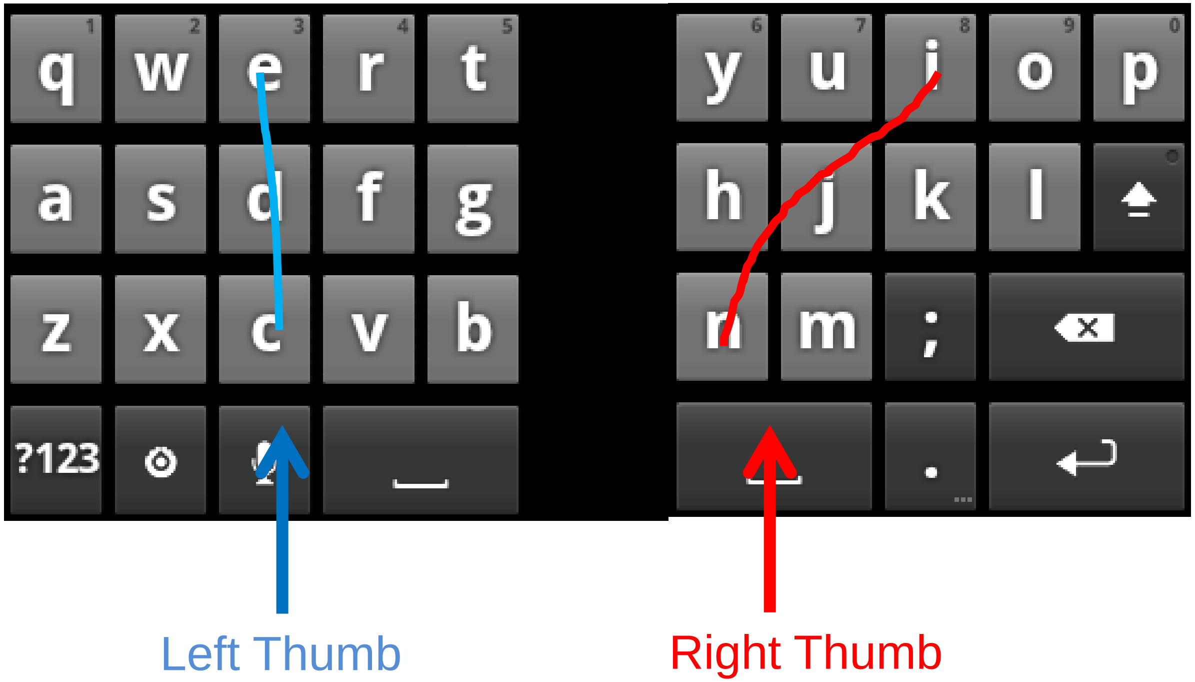
Swype

TouchPal


Search Results
Boston, MA

Google
national Parks

Bimanual Gesture Typing

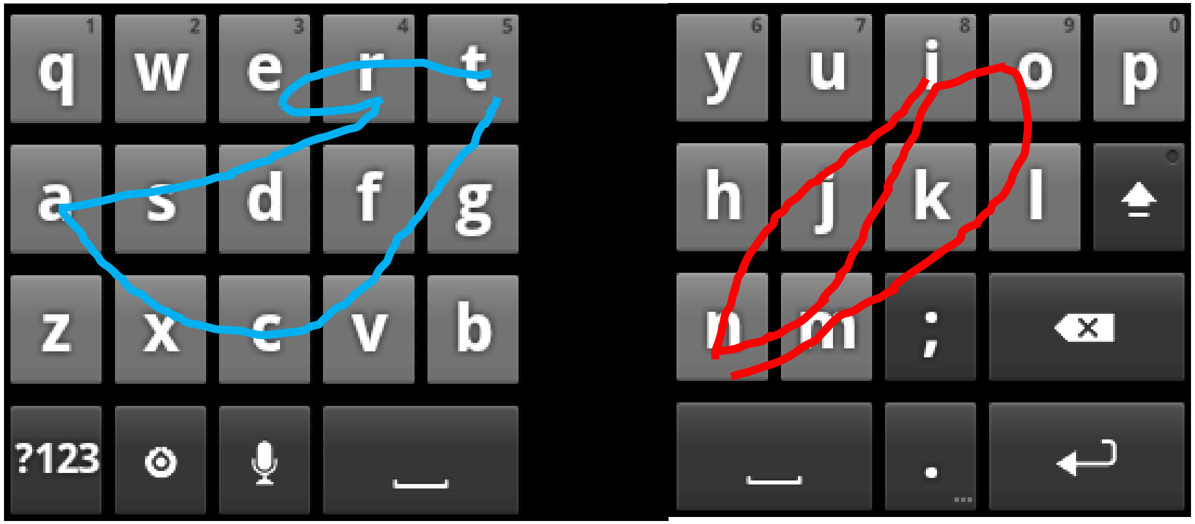
Entering *nice*



[**Bi,** Chelba, Ouyang, Partridge, Zhai. *Bimanual Gesture Keyboard.* ACM UIST2012]

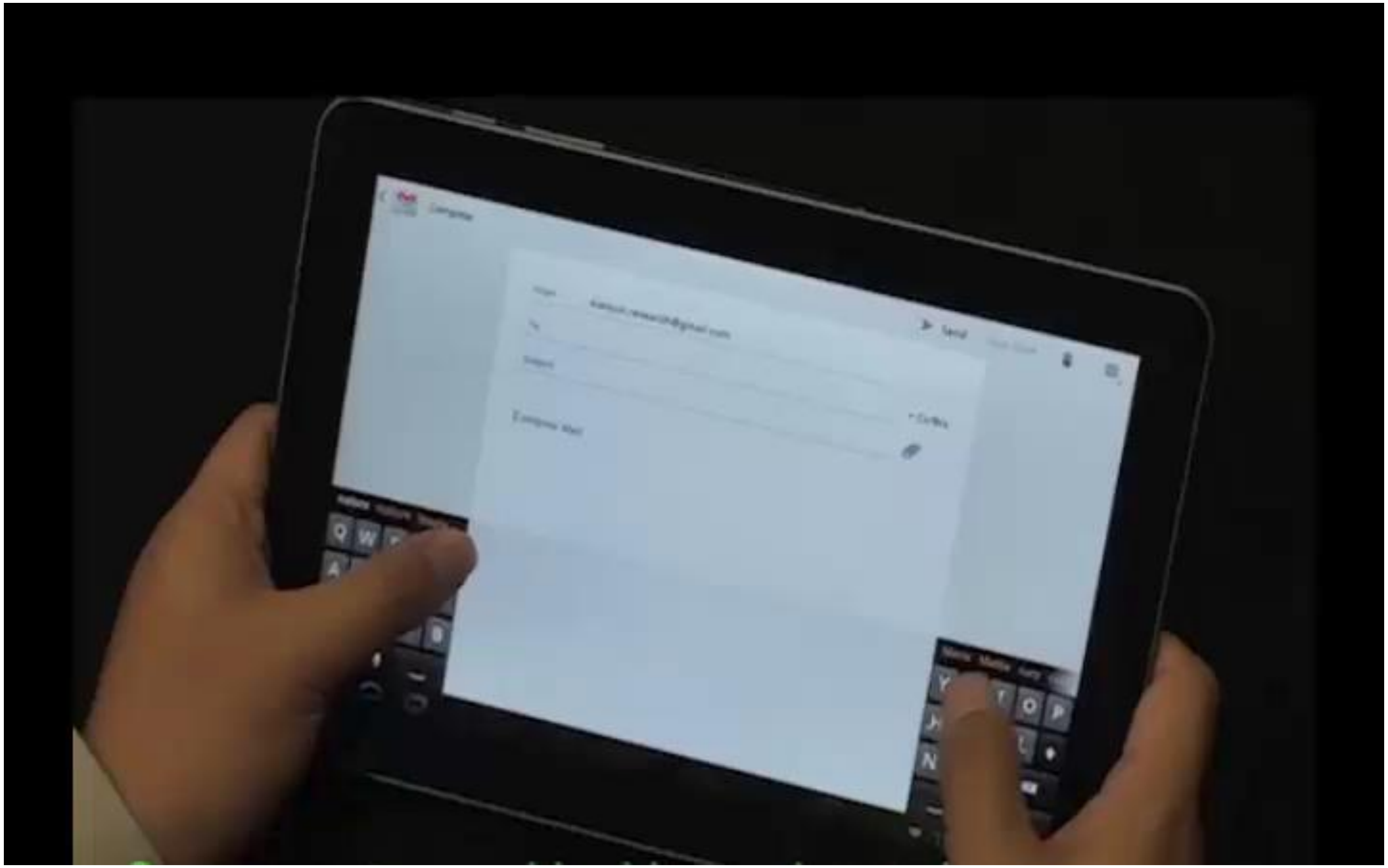
Bimanual Gesture Typing

Entering *interaction*



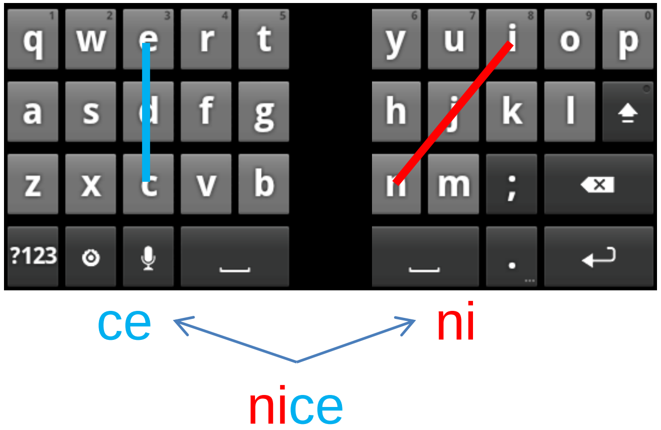
[**Bi,** Chelba, Ouyang, Partridge, Zhai. *Bimanual Gesture Keyboard.* ACM UIST2012]

Bimanual Gesture Typing



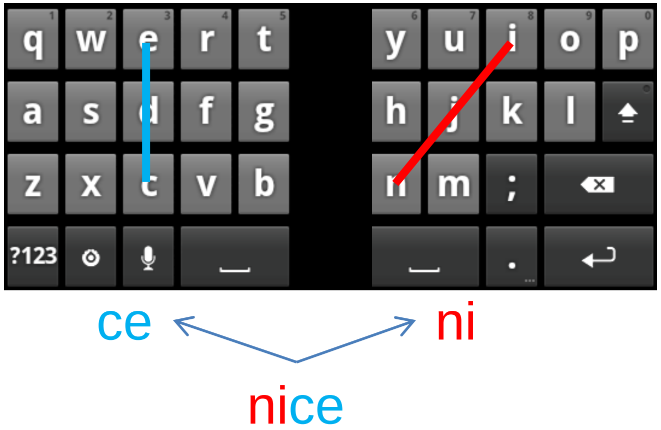
[**Bi,** Chelba, Ouyang, Partridge, Zhai. *Bimanual Gesture Keyboard.* ACM UIST2012]

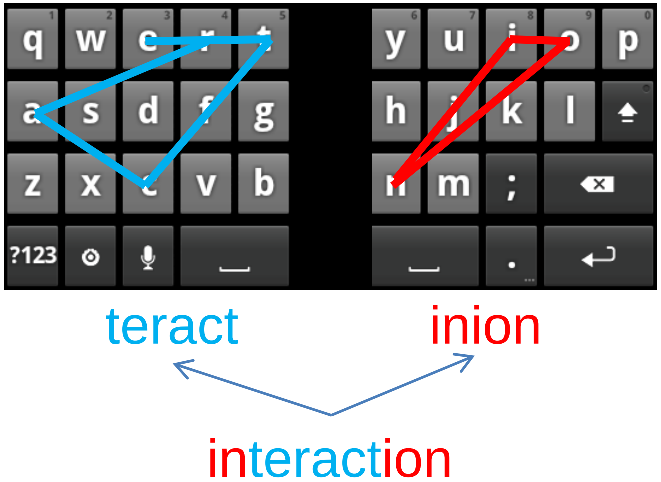
Perfect Templates



[**Bi,** Chelba, Ouyang, Partridge, Zhai. *Bimanual Gesture Keyboard.* ACM UIST2012]

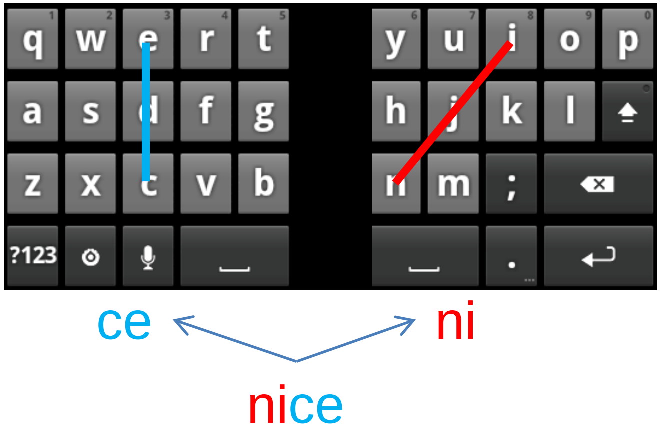
Perfect Templates

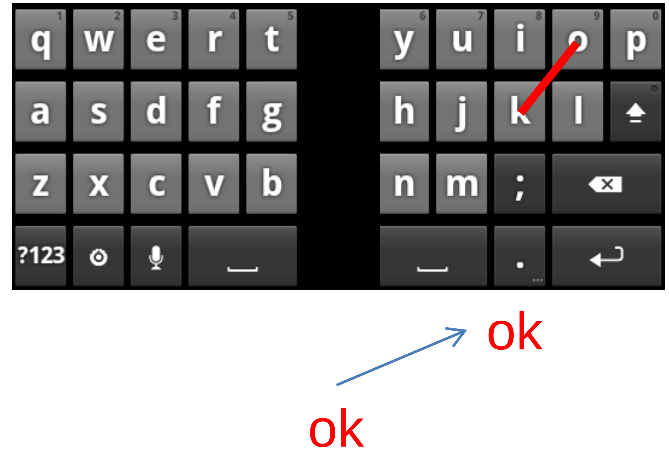


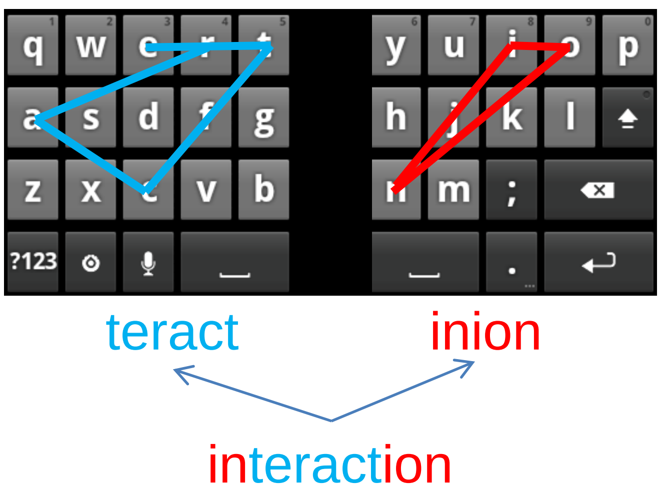


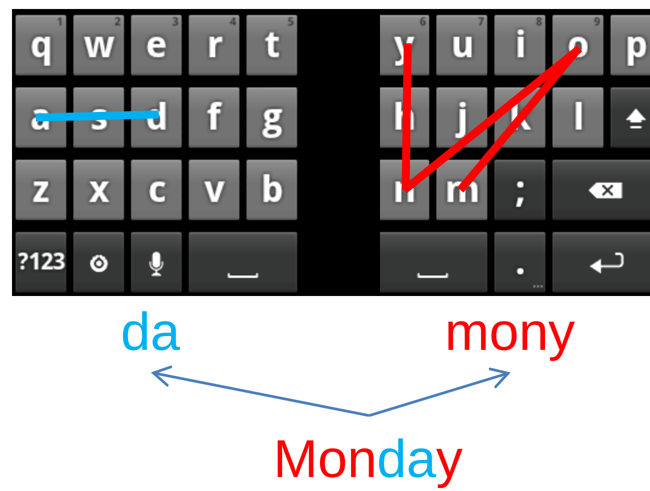
[**Bi,** Chelba, Ouyang, Partridge, Zhai. *Bimanual Gesture Keyboard.* ACM UIST2012]

Perfect Templates



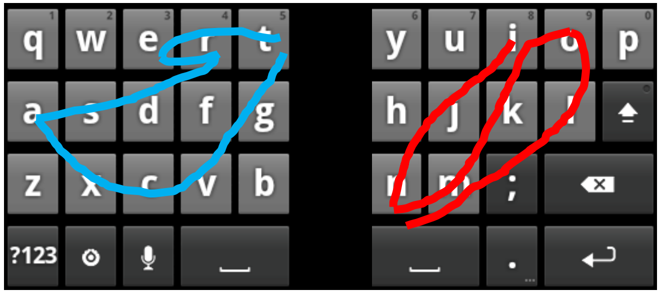






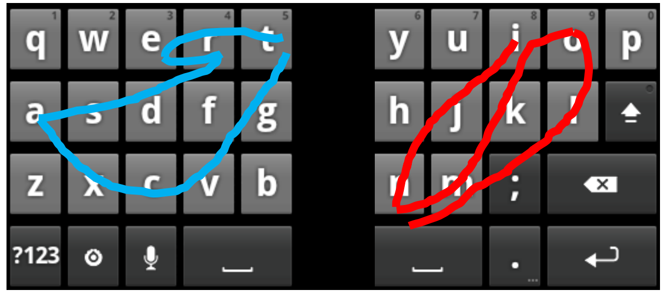
[**Bi,** Chelba, Ouyang, Partridge, Zhai. *Bimanual Gesture Keyboard.* ACM UIST2012]

Bimanual Gesture Recognition

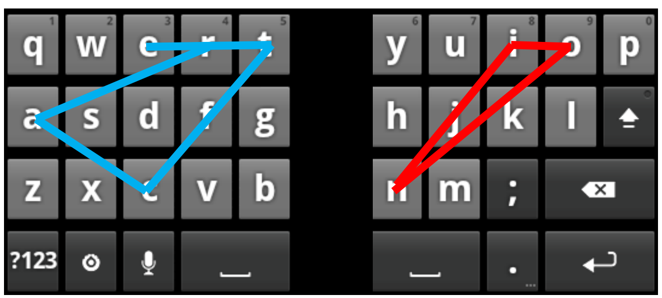


[**Bi,** Chelba, Ouyang, Partridge, Zhai. *Bimanual Gesture Keyboard.* ACM UIST2012]

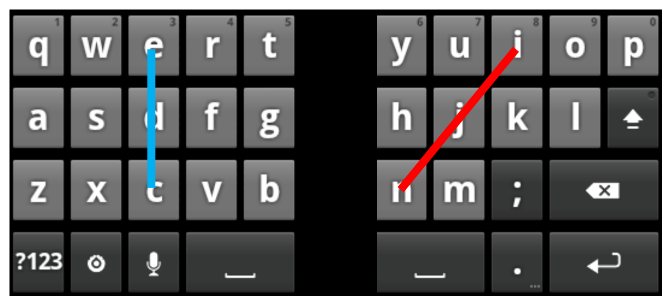
Bimanual Gesture Recognition



Best Match



interaction



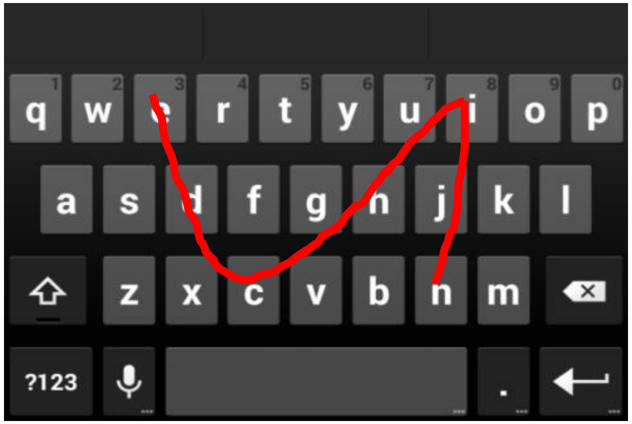
nice

…

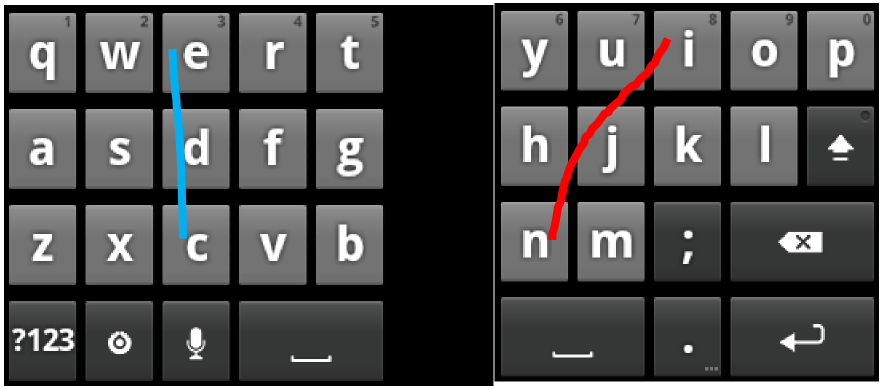
[**Bi,** Chelba, Ouyang, Partridge, Zhai. *Bimanual Gesture Keyboard.* ACM UIST2012]

Entering *nice*

Unimanual Gesture

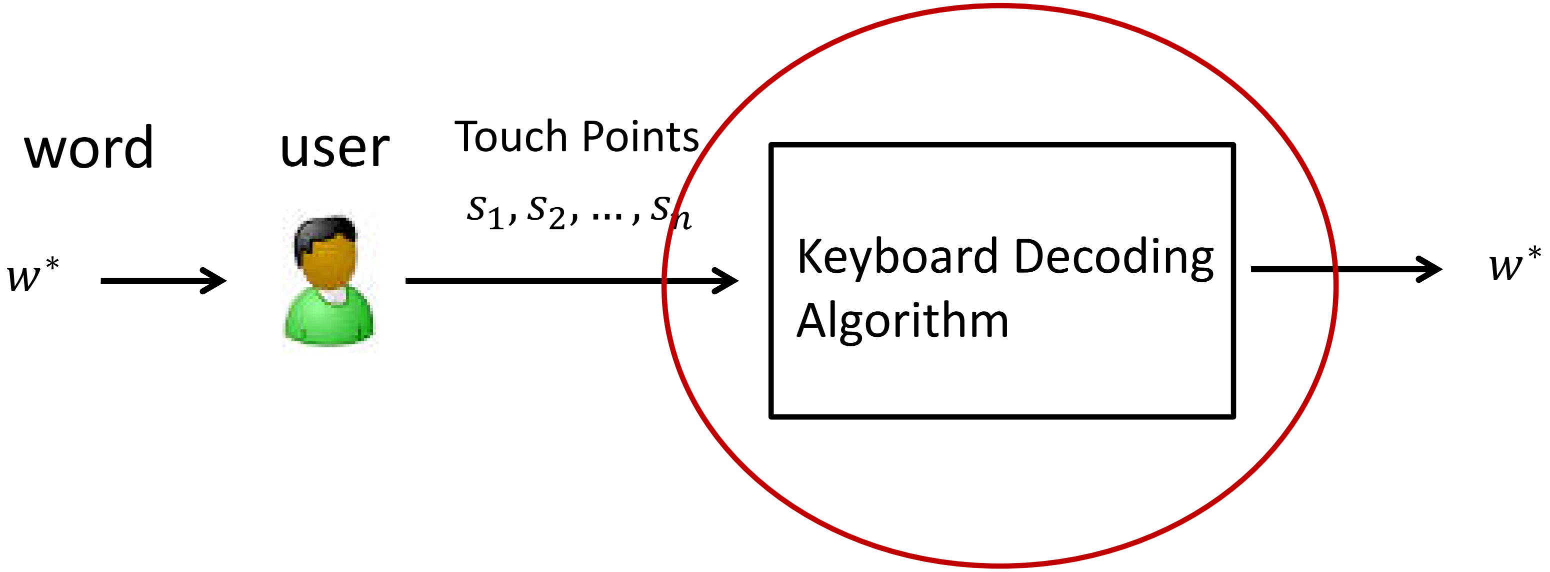


Bimanual Gesture

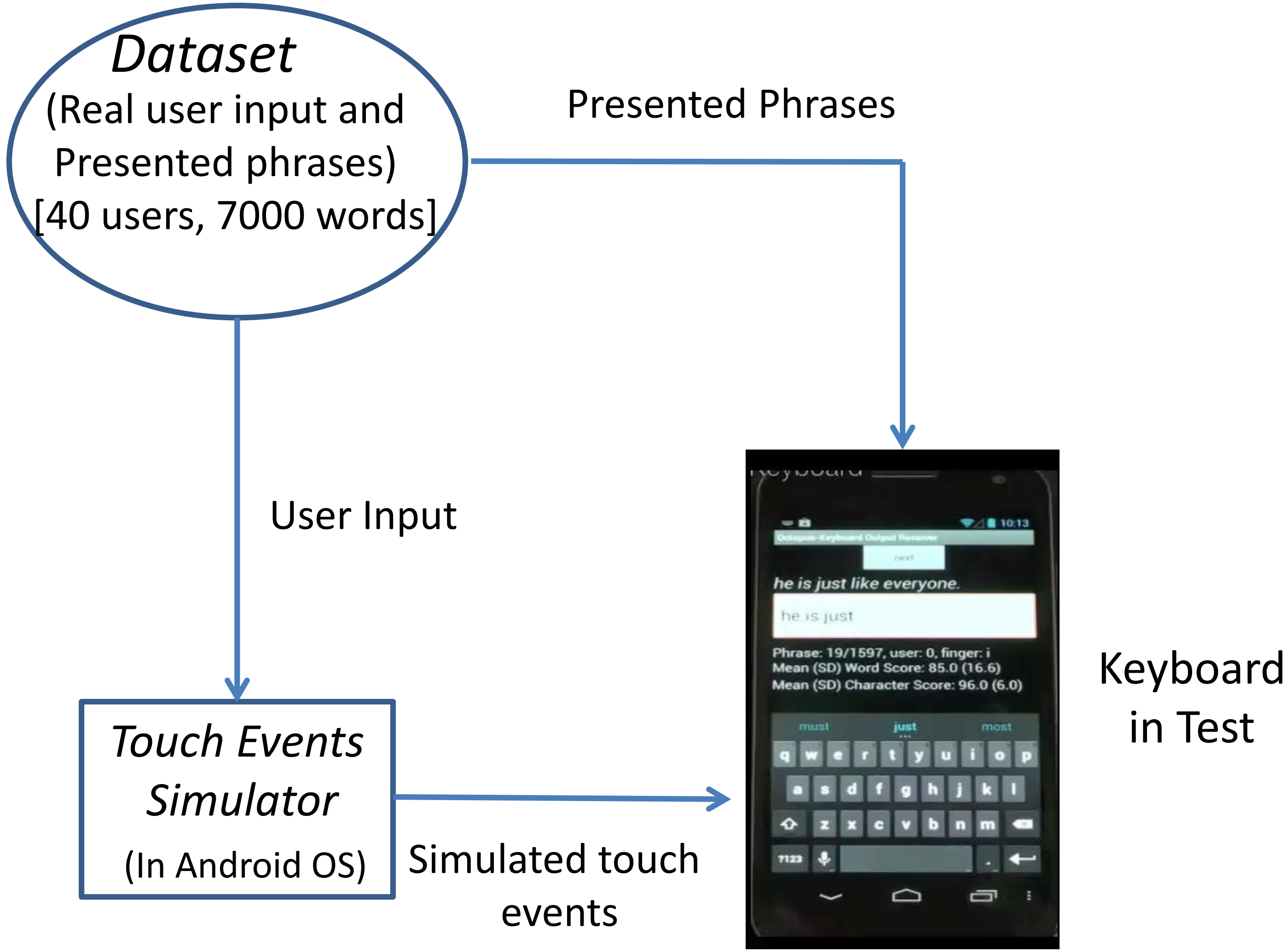


[**Bi,** Chelba, Ouyang, Partridge, Zhai. *Bimanual Gesture Keyboard.* ACM UIST2012]

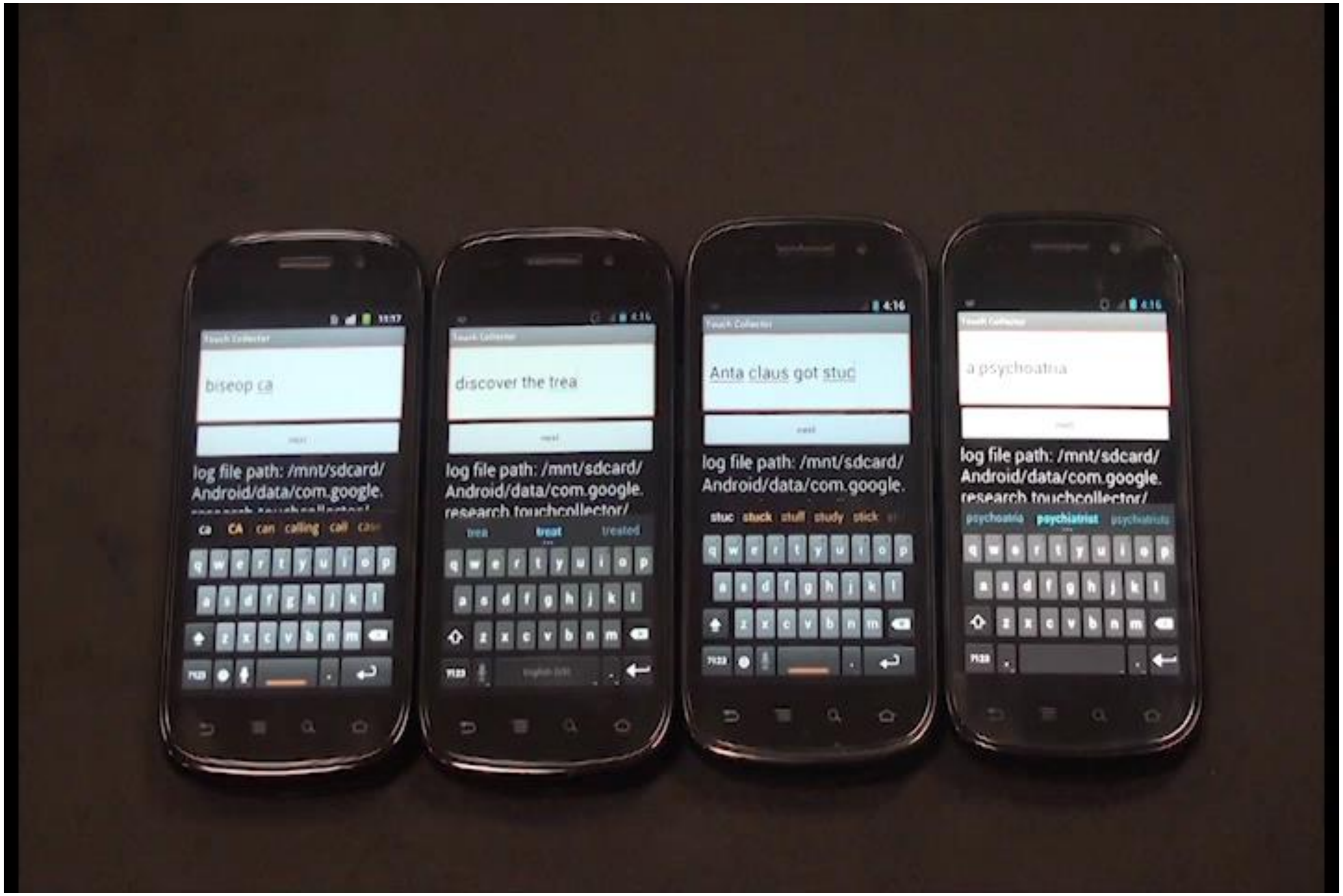
Optimization and Evaluation of Decoding Algorithm



Remulation: Replicating prior user study data with simulation.



[**Bi,** Azenkot, Partridge, Zhai. *Octopus: Evaluating Touchscreen Keyboard Correction and Recognition Algorithms via Remulation.* ACM CHI2013]



[**Bi,** Azenkot, Partridge, Zhai. *Octopus: Evaluating Touchscreen Keyboard Correction and Recognition Algorithms via Remulation.* ACM CHI2013]

Smart Touchscreen Keyboard

Correction

Thaml → Thank

$$Success\ Rate(W) = \frac{Correct\ Words}{Total\ Words}$$

[**Bi,** Ouyang, Zhai. *Both Complete and Correct? Multi-Objective Optimization of Touchscreen Keyboard.* ACM CHI2014]

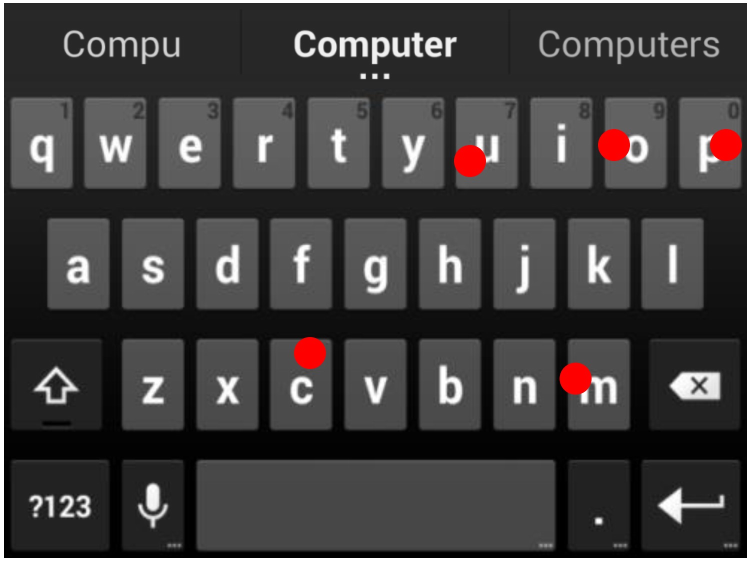
Smart Touchscreen Keyboard

Correction

Thaml → Thank

Completion

Compu → Computer



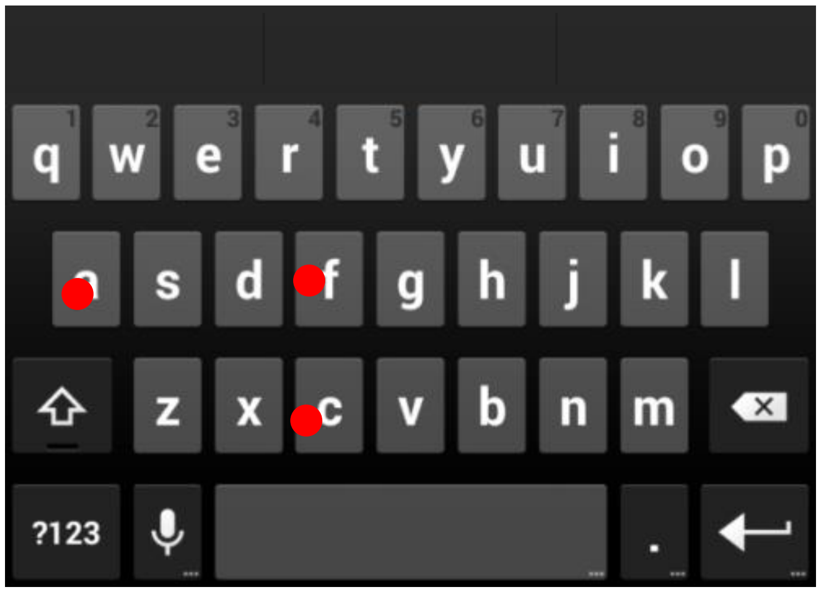
$$Success\ Rate(W) = \frac{Correct\ Words}{Total\ Words}$$

$$Keystroke\ Saving(S) = \frac{Saved\ Keystrokes}{Maximum\ Keystrokes}$$

[**Bi,** Ouyang, Zhai. *Both Complete and Correct? Multi-Objective Optimization of Touchscreen Keyboard.* ACM CHI2014]

Correction *vs.* Completion

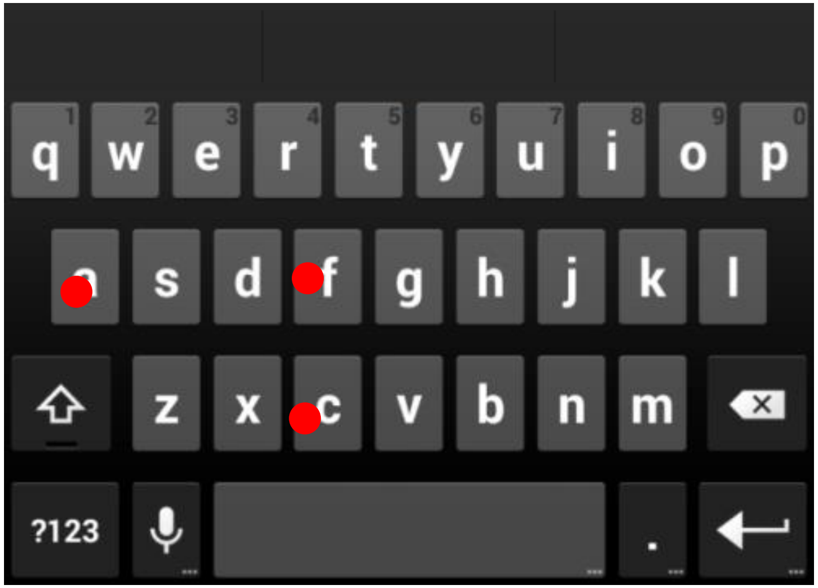
fac →



[**Bi,** Ouyang, Zhai. *Both Complete and Correct? Multi-Objective Optimization of Touchscreen Keyboard.* ACM CHI2014]

Correction *vs.* Completion

fac

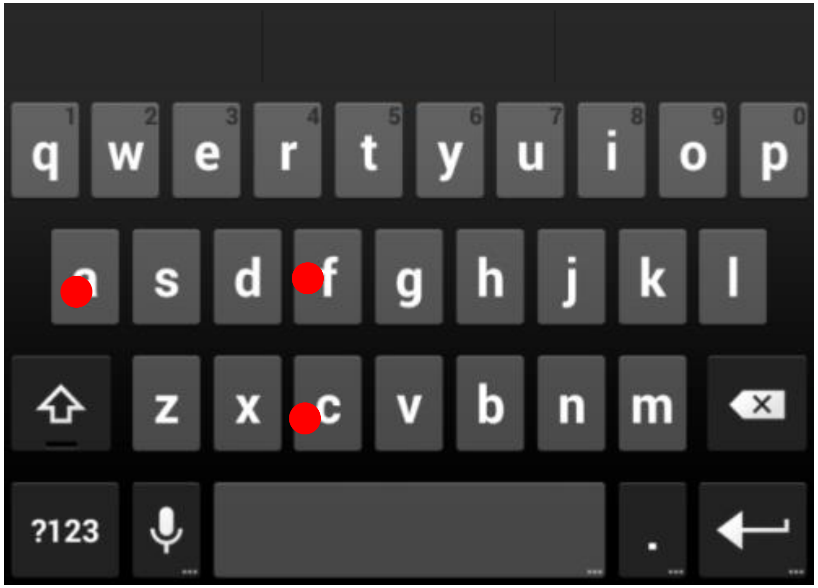


fax

[**Bi,** Ouyang, Zhai. *Both Complete and Correct? Multi-Objective Optimization of Touchscreen Keyboard.* ACM CHI2014]

Correction *vs.* Completion

fac



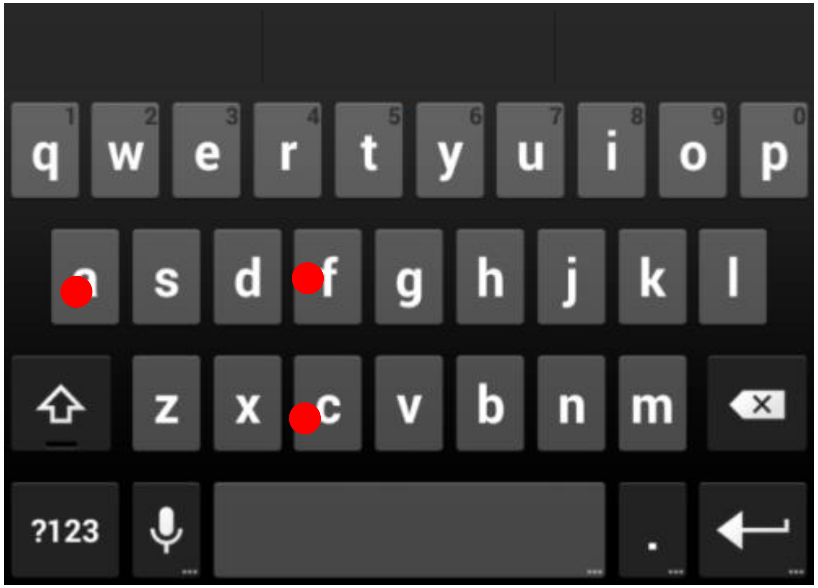
fax

facilitate

[**Bi,** Ouyang, Zhai. *Both Complete and Correct? Multi-Objective Optimization of Touchscreen Keyboard.* ACM CHI2014]

Correction *vs.* Completion

fac



fax

facilitate

Relationship between Correction and Completion ?

[**Bi,** Ouyang, Zhai. *Both Complete and Correct? Multi-Objective Optimization of Touchscreen Keyboard.* ACM CHI2014]

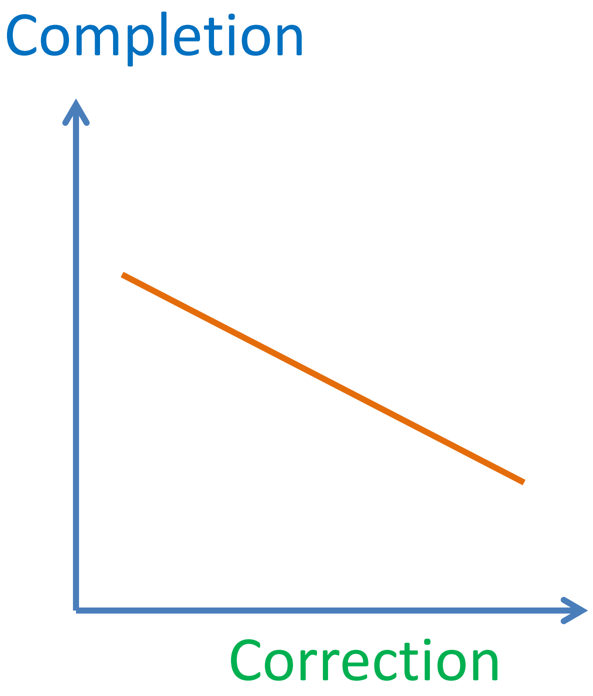
Correction *vs.* Completion

Possible Relationships

[**Bi,** Ouyang, Zhai. *Both Complete and Correct? Multi-Objective Optimization of Touchscreen Keyboard.* ACM CHI2014]

Correction *vs.* Completion

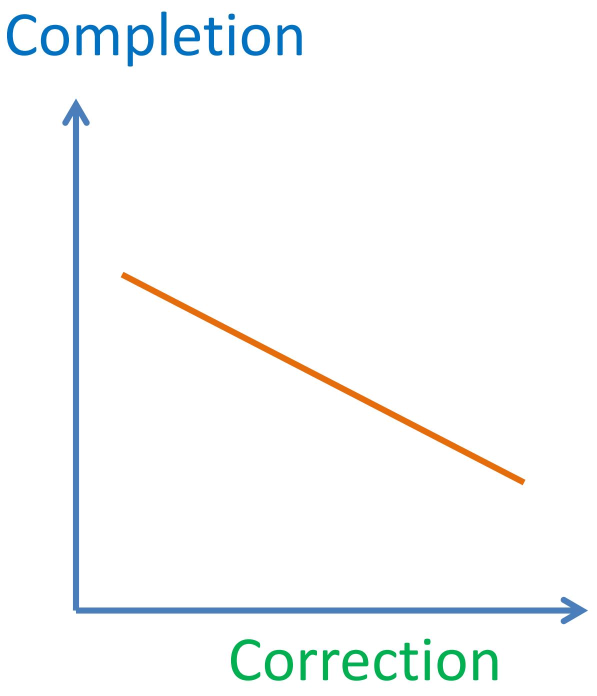
Possible Relationships

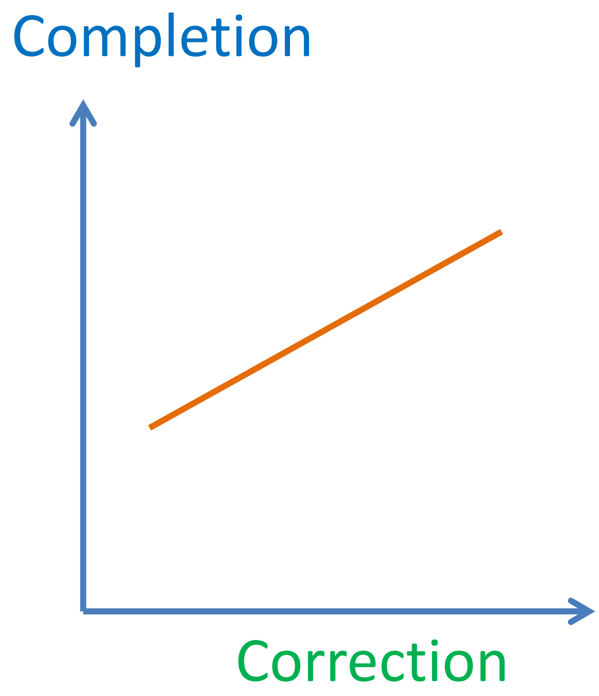


[**Bi,** Ouyang, Zhai. *Both Complete and Correct? Multi-Objective Optimization of Touchscreen Keyboard.* ACM CHI2014]

Correction *vs.* Completion

Possible Relationships

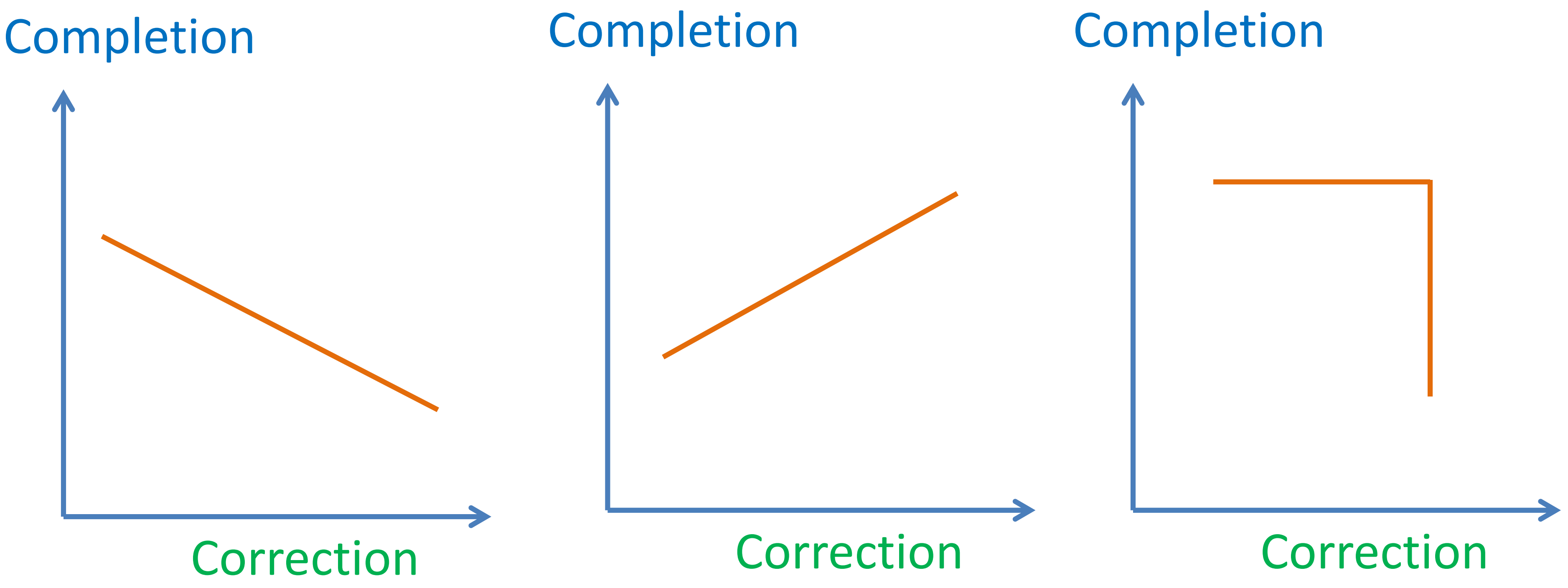




[**Bi,** Ouyang, Zhai. *Both Complete and Correct? Multi-Objective Optimization of Touchscreen Keyboard.* ACM CHI2014]

Correction *vs.* Completion

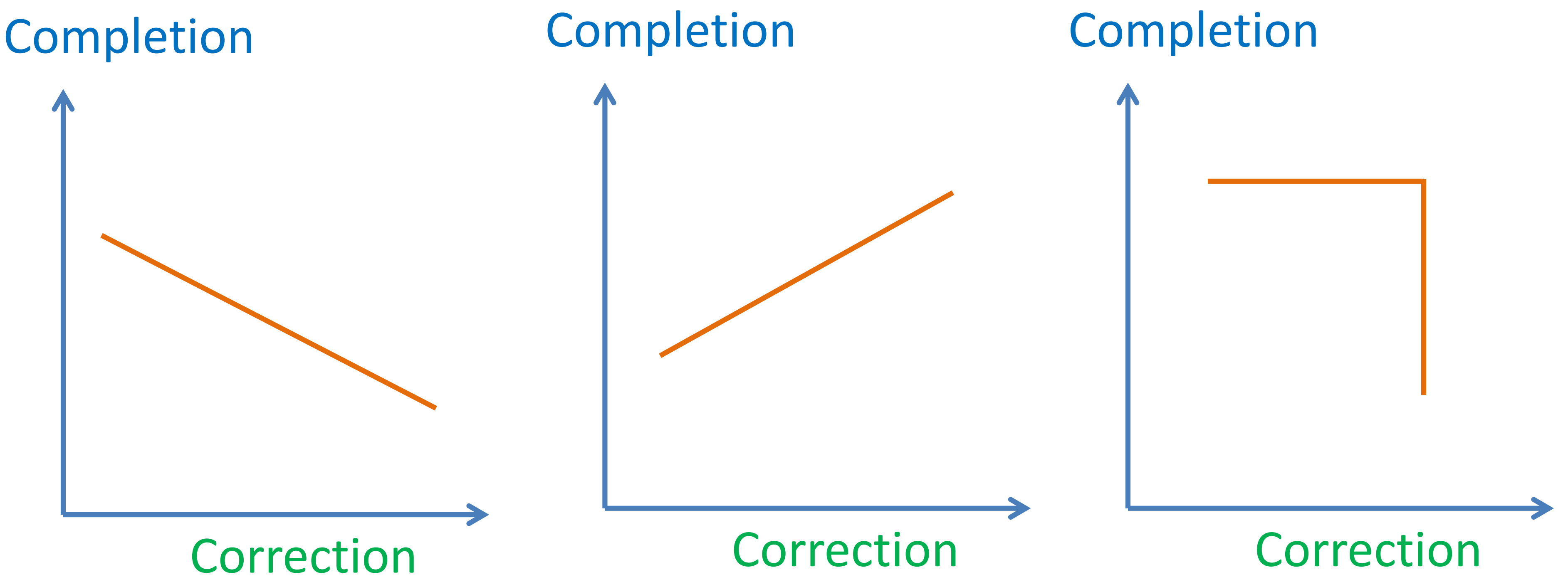
Possible Relationships



[**Bi,** Ouyang, Zhai. *Both Complete and Correct? Multi-Objective Optimization of Touchscreen Keyboard.* ACM CHI2014]

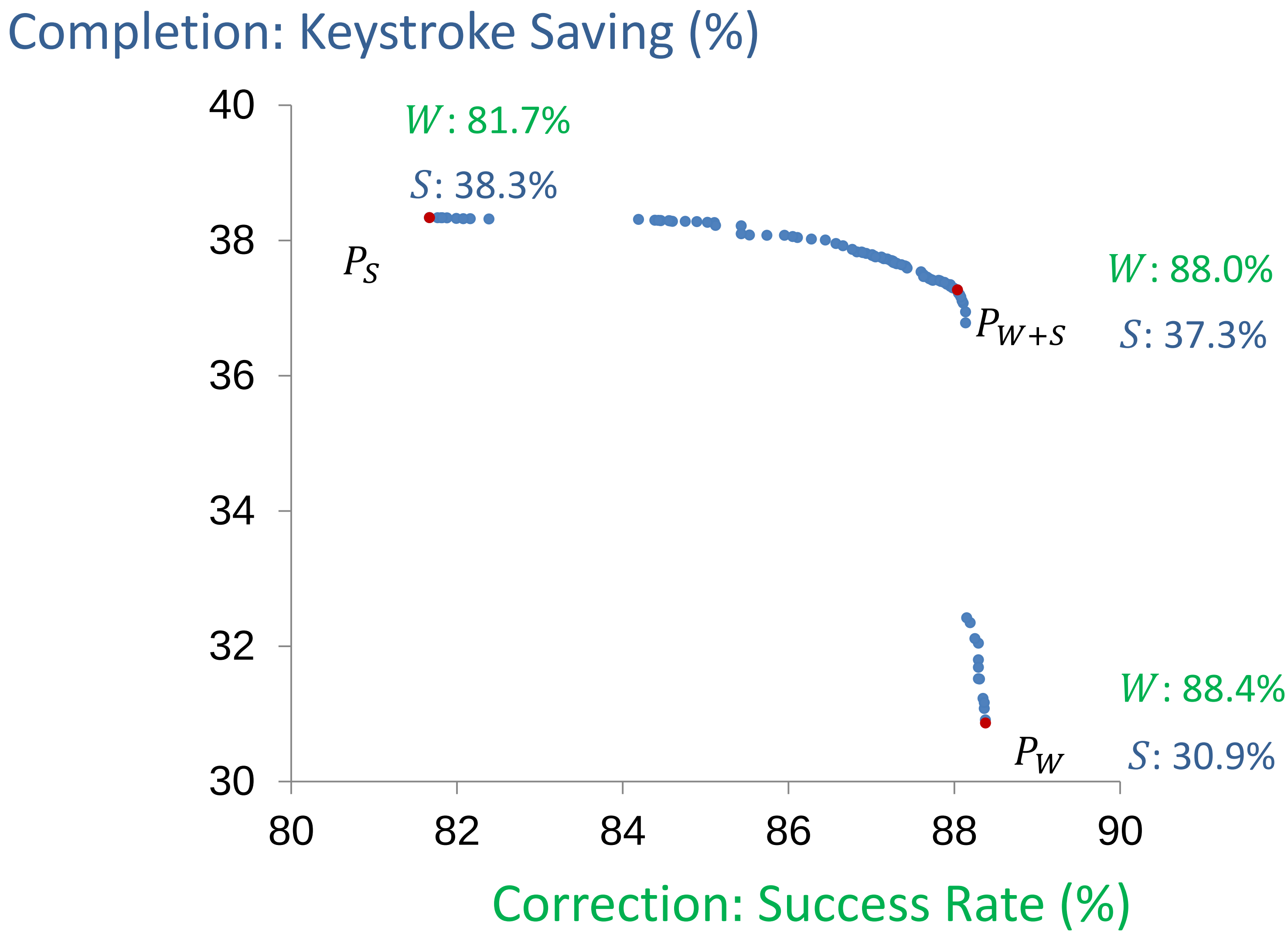
Correction *vs.* Completion

Possible Relationships ?



[**Bi,** Ouyang, Zhai. *Both Complete and Correct? Multi-Objective Optimization of Touchscreen Keyboard.* ACM CHI2014]

Pareto Frontier



[**Bi,** Ouyang, Zhai. *Both Complete and Correct? Multi-Objective Optimization of Touchscreen Keyboard.* ACM CHI2014]

Outline

- Smart Touch Keyboard
- Gesture Typing
- Optimizing Keyboard Layouts

Qwerty Layout

Q	W	E	R	T	Y	U	I	O	P
A	S	D	F	G	H	J	K	L	
	Z	X	C	V	B	N	M		

Qwerty is inefficient for one finger typing.

Optimization Objective Function

- Fitts' Law (Fitts 1954):

$$MT_{ij} = a + b\log_2\left(\frac{D_{ij}}{W} + 1\right)$$

MT_{ij}: Movement Time from Key *i* to Key *j*
D_{ij} : Distance from Key *i* to Key *j*
W *:* Key Width

Optimization Objective Function

- Fitts' Law (Fitts 1954):

$$MT_{ij} = a + b\log_2\left(\frac{D_{ij}}{W} + 1\right)$$

MT_{ij}: Movement Time from Key *i* to Key *j*
D_{ij} : Distance from Key *i* to Key *j*
W *:* Key Width

- Average time of typing a letter:

$$t = a + b\sum_{i}^{26}\sum_{j}^{26} P_{ij}\log_2\left(\frac{D_{ij}}{W} + 1\right)$$

P_{ij}: Frequency of an ordered letter pair *i, j*

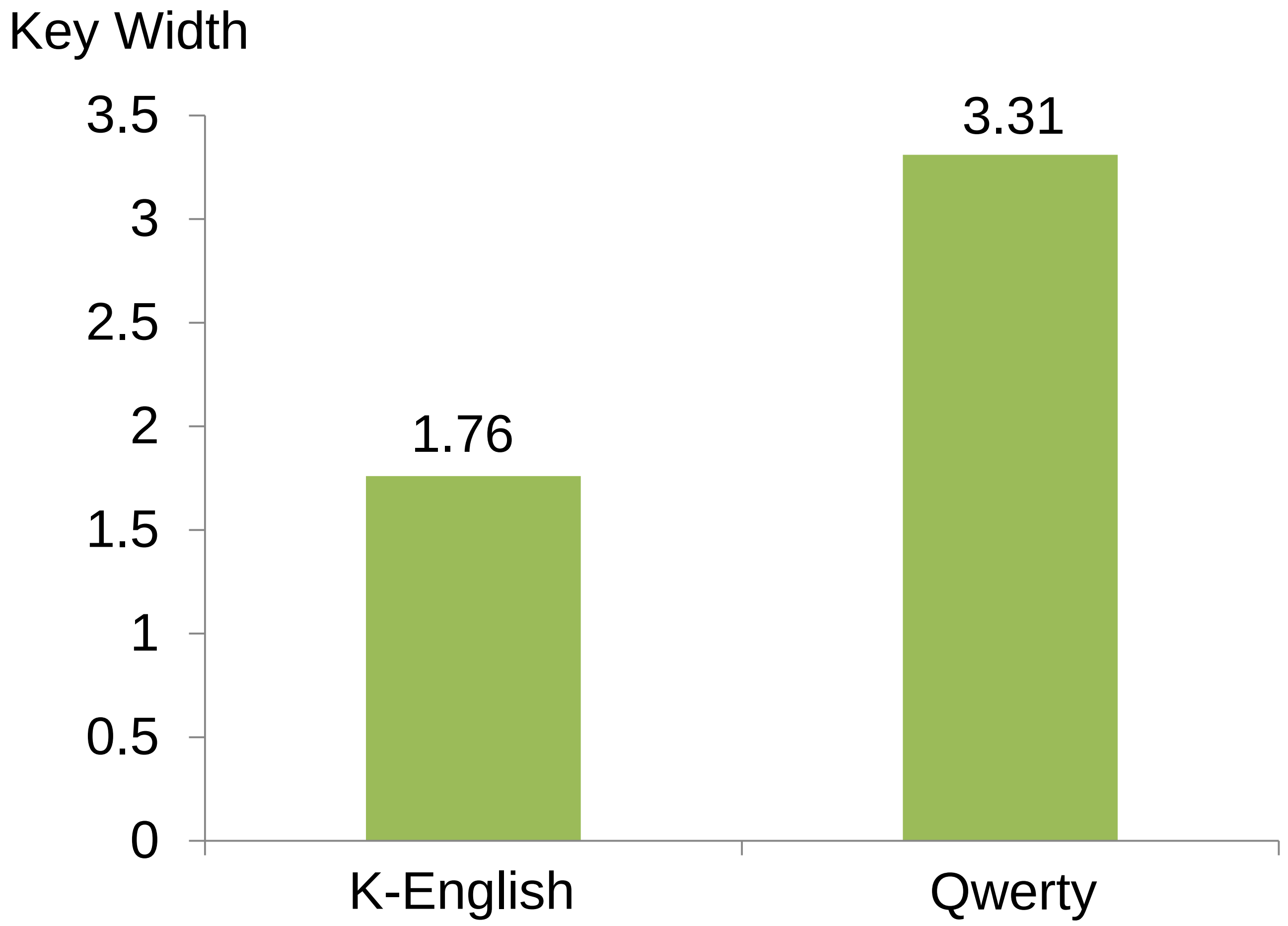
Layout Optimized for English

K-English

Z	J	D	G	K	
Y	L	N	I	C	
F	O	A	T	H	W
B	U	R	E	S	
Q	P	M	V	X	

[**Bi,** Smith, Zhai. *Multilingual Touchscreen Keyboard Design and Optimization* Human-Computer Interaction 2012]

Average Finger Travel Distance



[**Bi,** Smith, Zhai. *Multilingual Touchscreen Keyboard Design and Optimization* Human-Computer Interaction 2012]

Layout Optimized for Five Languages

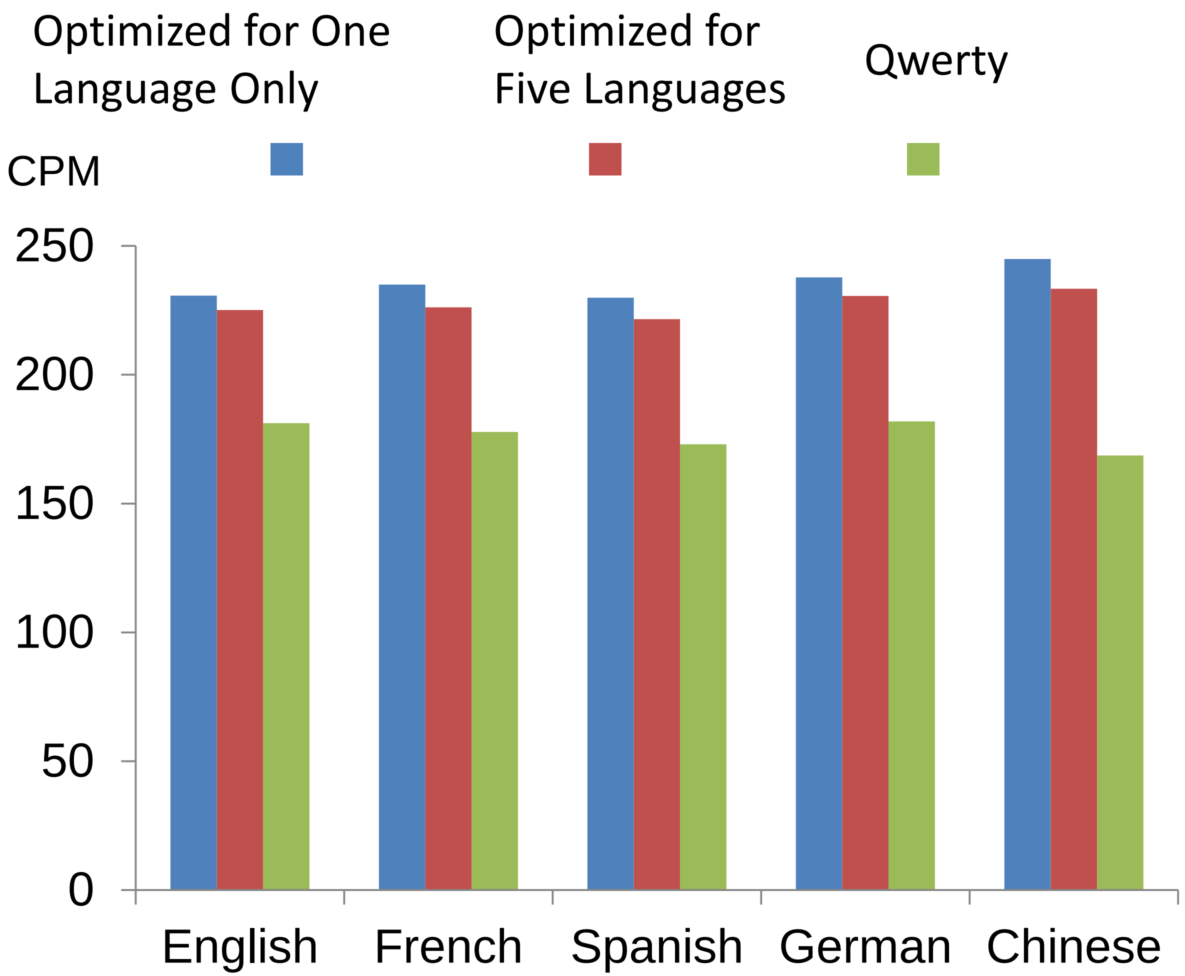
English, French, German, Spanish, and Chinese *Pinyin*

K5

	K	J	Z	X	
	F	C	H	T	W
Q	U	O	I	S	P
Y	M	A	N	E	R
	B	L	G	D	V

[**Bi,** Smith, Zhai. *Multilingual Touchscreen Keyboard Design and Optimization* Human-Computer Interaction 2012]

Typing Speed



[**Bi,** Smith, Zhai. *Multilingual Touchscreen Keyboard Design and Optimization* Human-Computer Interaction 2012]

Optimized Layout for Gesture Typing

Qwerty

Q	W	E	R	T	Y	U	I	O	P
A	S	D	F	G	H	J	K	L	
	Z	X	C	V	B	N	M		

[Smith, **Bi,** Zhai. *Optimizing Touchscreen Keyboard for Gesture Typing,* CHI2015, 3365-3374]

Optimized Layout for Gesture Typing

or vs. our

Qwerty

Q	W	E	R	T	Y	U	I	O	P
A	S	D	F	G	H	J	K	L	
	Z	X	C	V	B	N	M		

GK-T (Speed, Clarity, Similarity to Qwerty)

Q	D	W	S	O	I	Y	U	J	P
Z	R	F	A	T	N	G	K	L	
	C	E	X	H	V	M	B		

[Smith, **Bi,** Zhai. *Optimizing Touchscreen Keyboard for Gesture Typing,* CHI2015, 3365-3374]

Optimized Layout for Gesture Typing

or vs. our

Qwerty

Q	W	E	R	T	Y	U	I	O	P
A	S	D	F	G	H	J	K	L	
	Z	X	C	V	B	N	M		

GK-T (Speed, Clarity, Similarity to Qwerty)

Q	D	W	S	O	I	Y	U	J	P
Z	R	F	A	T	N	G	K	L	
	C	E	X	H	V	M	B		

[Smith, **Bi,** Zhai. *Optimizing Touchscreen Keyboard for Gesture Typing,* CHI2015, 3365-3374]

COMPASS: Rotational Keyboard on Non-Touch Smartwatches

Xin Yi, Chun Yu, Weijie Xu, Xiaojun Bi, and Yuanchun Shi. 2017. COMPASS: Rotational Keyboard on Non-Touch Smartwatches. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17). ACM, New York, NY, USA, 705-715.

Smartwatches are popular

But what if…

COMPASS

- Circular Keyboard
- Multi-cursor paradigm
- Bezel rotation
- Physical button
- Dynamic Cursor Placement
- Flick to delete


hello_
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Multi-Cursor Paradigm

1- Cursor

$dis('M') = 11$

3-Cursors

$dis('M') = 4$

Algorithm of COMPASS

Similar as an ambiguous keyboard

Candidate word

Users' input

$$P(\boxed{W}|\boxed{I}) \propto \boxed{P(I|W)} \times \boxed{P(W)}$$

$$P(I|W) = \begin{cases} 1 \; if \; I \; matches \; with \; W \\ 0 \; otherwise \end{cases}$$

Top 15,000 words and frequencies in American National Corpus

Visual Cues

1. For each character c, calculate $P(c)$ as the probability of it being the next character given the input prefix.
2. Each key is highlighted according to $P(c)$.
3. Impossible characters are dimmed.

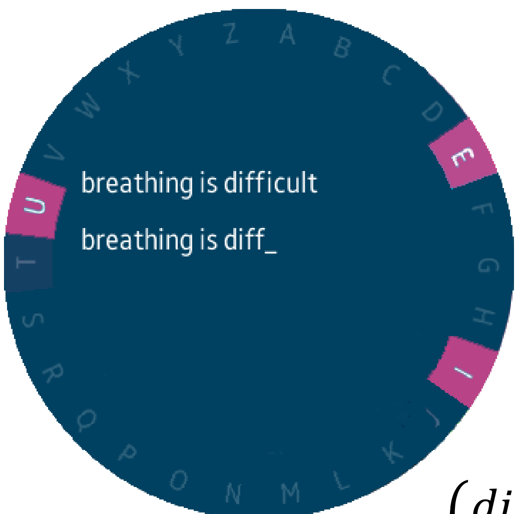
Cursor Adjustment

Locations of cursors dynamically adjust upon each character selection.

Choose the locations that would minimize the *Expected Next Rotation Distance*

$$ENRD = \sum_{c \in \chi} dis(c) \times P(c)$$

“Bad Case”



$$\begin{cases} dis(E) = 0 \\ dis(I) = 0 \\ dis(U) = 0 \end{cases}$$

Naïve Optimization

$$\#possible\ next\ key = 5$$

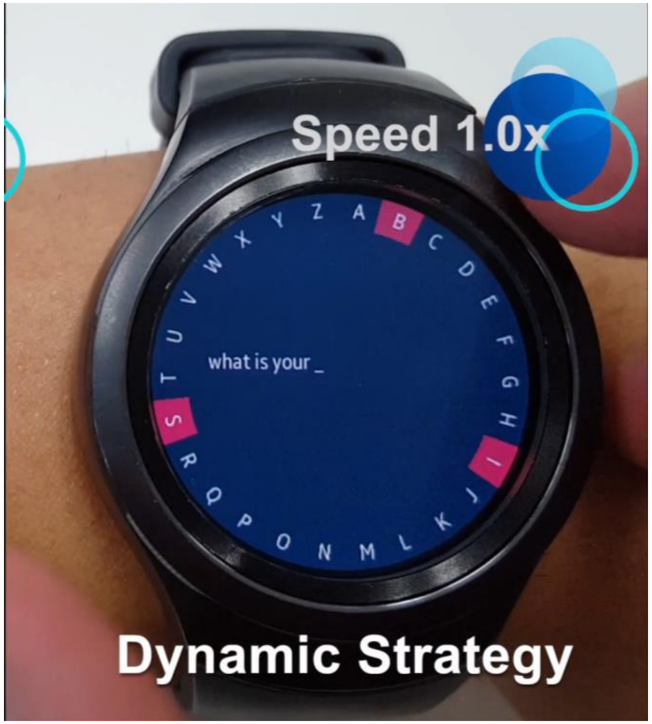
$$\begin{cases} dis(E) = 0 \\ dis(I) = 1 \\ dis(U) = 1 \end{cases}$$

Restriction Applied

$$\#possible\ next\ key = 1$$

S-COMPASS & D-COMPASS

S(Static)-COMPASS



D(Dynamic)-COMPASS

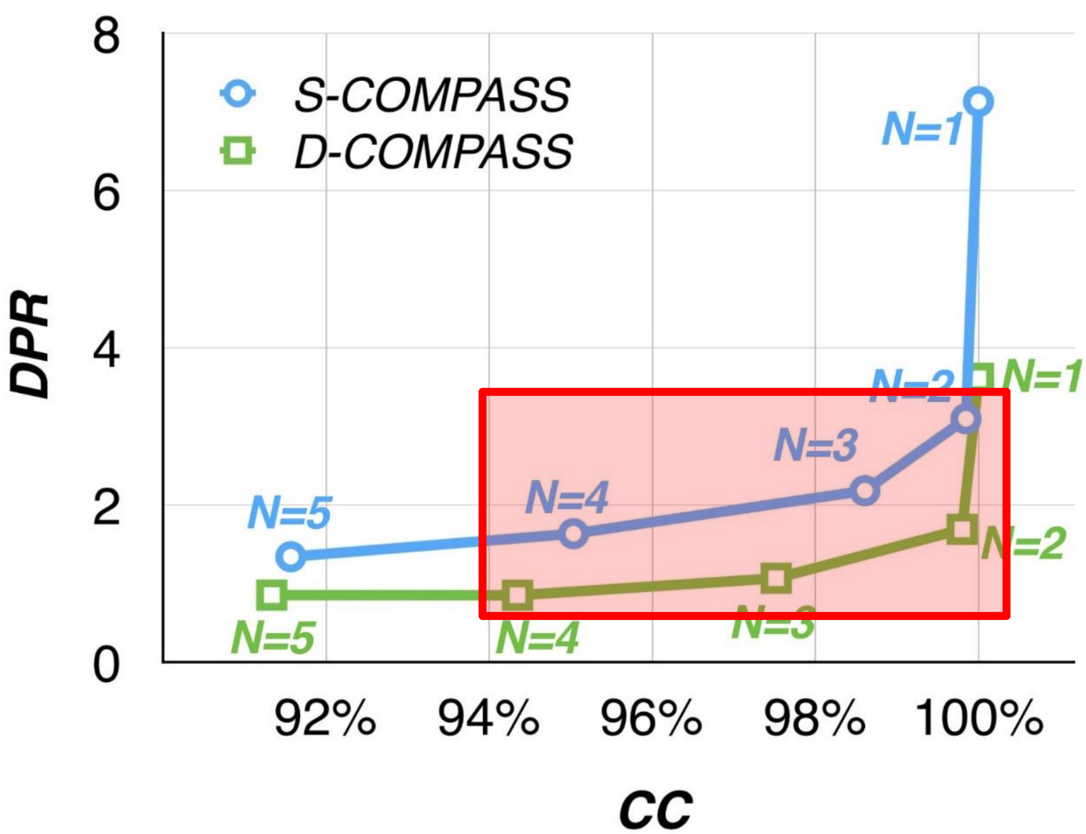
Number of Cursors *(N)*

- Simulated all 15,000 words in the corpus
- Assumed perfect user input
- $1 \leq N \leq 5$

DPR (Distance Per Rotation): Average distance of each rotation

CC (Candidate Coverage): Ratio of words that appear in the top-3 candidates given users' perfect input

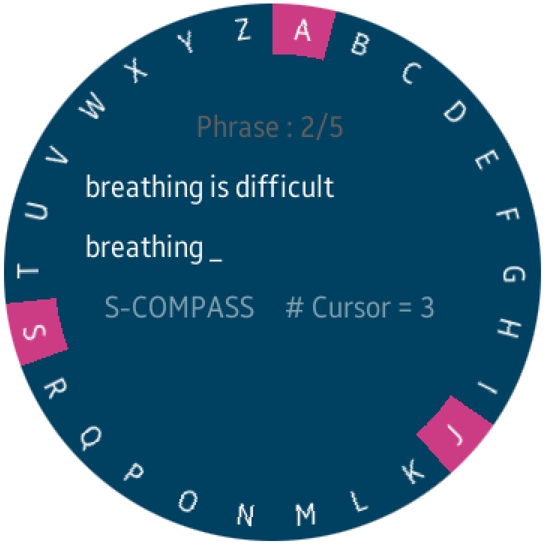
Simulation Result



DPR (Distance Per Rotation): Average distance of each rotation

CC (Candidate Coverage): Ratio of words that appear in the top-3 candidates given users' perfect input

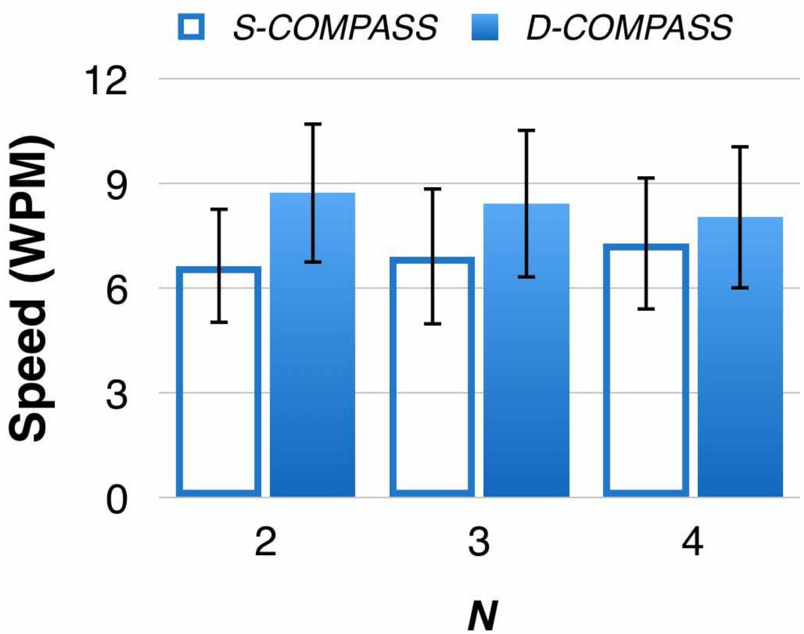
User Study



12 participants
× 2 techniques (*S-COMPASS* and *D-COMPASS*)
× 3 N (2, 3 and 4)
× 5 phrases


what is your _
Static Strategy
what is your _
Dynamic Strategy

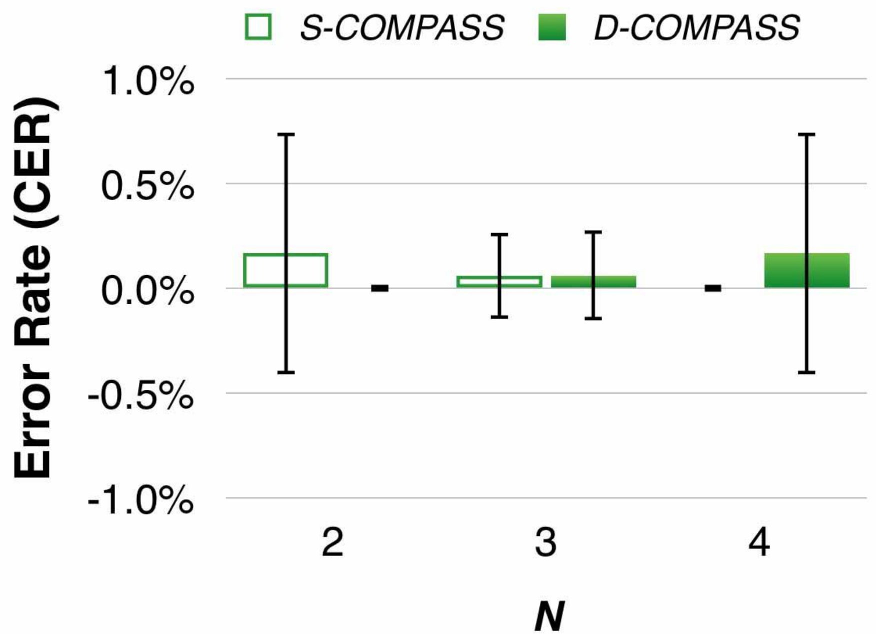
Speed & Error Rate



N – Speed

$\boldsymbol{F_{2,22} = 3.42, p = .05}$ (*S-COMPASS*)

$\boldsymbol{F_{2,22} = 7.04, p < .01}$ (*D-COMPASS*)



N – Error Rate

$\boldsymbol{F_{2,22} = 0.66, n.s.}$ (*S-COMPASS*)

$\boldsymbol{F_{2,22} = 1.00, n.s.}$ (*D-COMPASS*)

Final Design

D-COMPASS

$N = 3$

Auto-completion

Auto-Completion

$$\mathrm{I} = I_1 I_2 \cdots I_n$$

$$\mathrm{W} = W_1 W_2 \cdots W_n W_{n+1} \cdots W_m$$

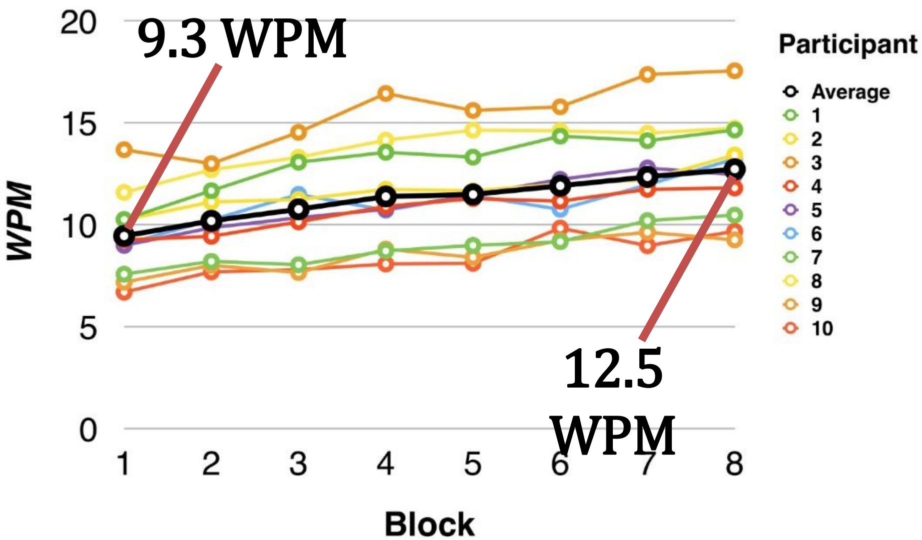
$$P(I|W) = \begin{cases} 1 \; if\ I\ matches\ with\ W \\ 0 \; otherwise \end{cases}$$

$$P(I|W) = \begin{cases} \alpha^{m-n} \; if\ I\ matches\ with\ the\ prefix\ of\ W \\ 0 \; otherwise \end{cases}$$

$\alpha = 0.7$ as the penalty of looking ahead

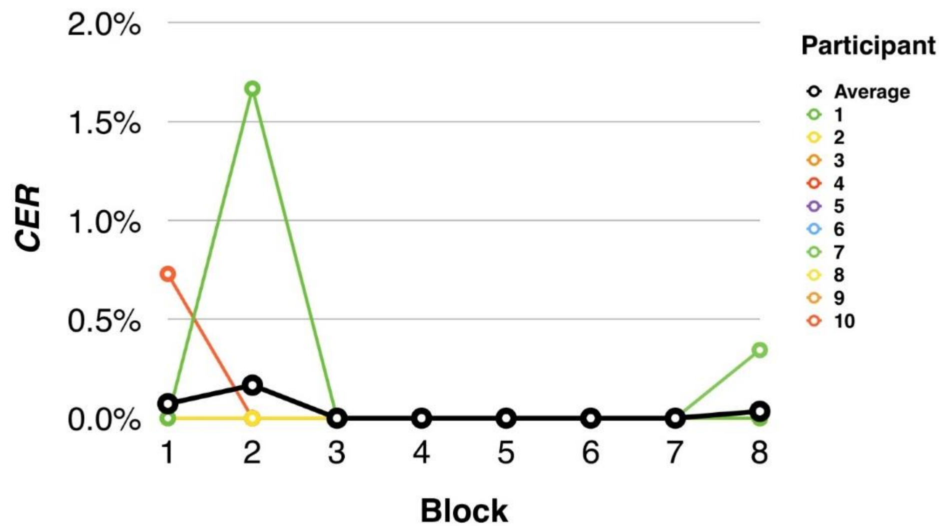
Speed & Error Rate

10 participants × 8 blocks × 10 phrases



Block – Speed

$F_{7,63} = 48.0, p < .0001$



Block – Error Rate

$F_{7,63} = 0.81, n.s.$

Applications