Intelligently Recommending Key Bindings on Physical Keyboards with Demonstrations in Emacs

Shudan Zhong[1]   Hong Xu[2]

sdz@berkeley.edu, hongx@usc.edu

March 18, 2019

[1]University of California, Berkeley

[2]University of Southern California

Executive Summary

- Key bindings are the main means to interact with a computer with a physical keyboard.
- Different key bindings can lead to drastically different user experience and productivity.
- Surprisingly, research on recommender systems for key bindings are lacking.
- We propose the first (to the best of our knowledge) such recommender system and demonstrate it in Emacs.

Agenda

- Key Bindings on Physical Keyboards

- The Emacs Key Binding Recommender System (EKBRS) as a Demonstration
  - Details of the Emacs Key Binding Recommender System (EKBRS)
  - Empirical Evaluation

- Conclusion

Agenda

- Key Bindings on Physical Keyboards

- The Emacs Key Binding Recommender System (EKBRS) as a Demonstration

  - Details of the Emacs Key Binding Recommender System (EKBRS)

  - Empirical Evaluation

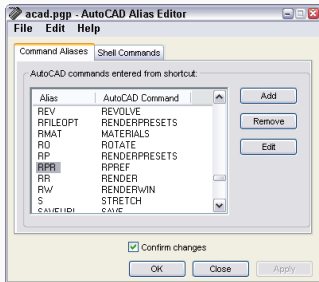- Conclusion

What Are Key Bindings?

- Key bindings define how users interact with computers via physical keyboards.
- (Classic) examples:
  - Copy: `Ctrl` + `c`
  - Paste: `Ctrl` + `v`
  - Save: `Ctrl` + `s`
- Applications define their own key bindings.

Key Bindings Are Important

Key bindings play essential roles in many professional applications.

(a) Programming

(b) Computer-Aided Design

(c) Professional Gaming

Image sources:

(a) `http://blog.markpearl.co.za/Ultimate-Developer-Keyboard`

(b) `https://allaboutcad.com/tutorial-create-a-command-alias-keyboard-shortcut/`

(c) `https://www.logitechg.com/en-us/products/gaming-mice/g600-mmo-gaming-mouse.html`

Key Bindings Are Important

Different key bindings can lead to drastically different productivity and user experience.

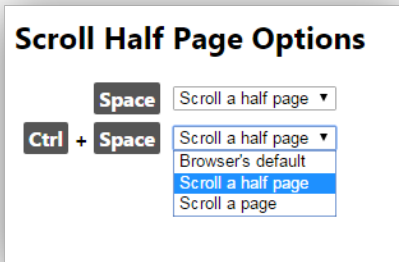- An absurd example: $\boxed{\uparrow}$ for moving down and $\boxed{\downarrow}$ for moving up.

Image source: https://i.redd.it/al15957ctaky.jpg

Key Bindings Are Important

Different key bindings can lead to drastically different user experience.

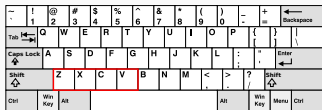- In browsers, [Space] is commonly used for scrolling one page down. Users can comfortably sit back and tap the spacious [Space].

Image source: `https:`

`//softwarerecs.stackexchange.com/questions/19804/browser-plugin-to-change-the-amount-of-scroll-on-space-key`
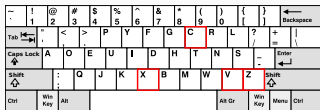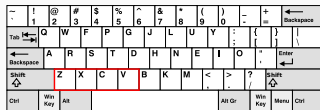
Different key bindings can lead to drastically different productivity and user experience.

(a) qwerty (b) Dvorak (c) Colemak

Ctrl + z , Ctrl + x , Ctrl + c , Ctrl + v

Image source:
(a) https://commons.wikimedia.org/wiki/File:KB_United_States.svg
(b) https://commons.wikimedia.org/wiki/File:KB_United_States_Dvorak.svg

(c) https://commons.wikimedia.org/wiki/File:KB_US-Colemak.svg

Intelligently Recommending Key Bindings

- Although key bindings are important, surprisingly, research on applying recommender systems to key bindings on physical keyboards is lacking.
- We propose the first (to the best of our knowledge) recommender system that recommends key bindings.
- We demonstrate such the recommender system in Emacs.

Agenda

- Key Bindings on Physical Keyboards

- The Emacs Key Binding Recommender System (EKBRS) as a Demonstration
  - Details of the Emacs Key Binding Recommender System (EKBRS)
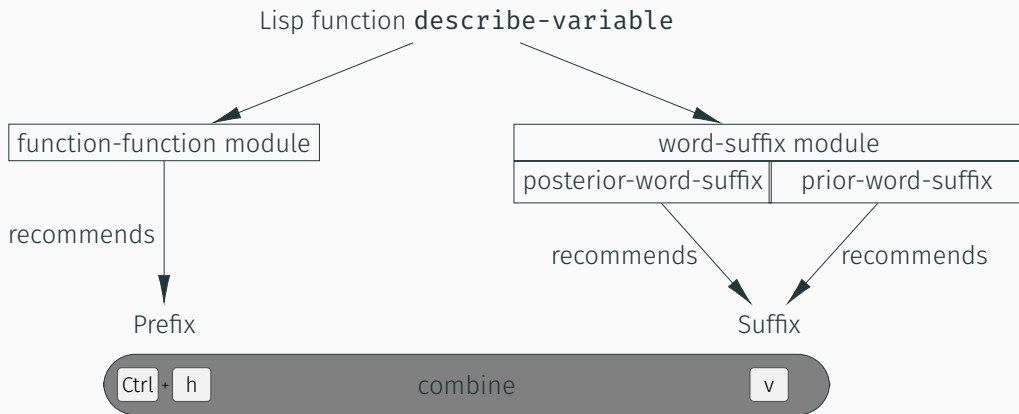 - Empirical Evaluation

- Conclusion

Why Demonstrate in Emacs?

- Key bindings matter in Emacs: It heavily exploits key bindings in its user interface.
- Extensive customizability:
  - Emacs is a Lisp machine (Lisp interpreter + Lisp library + user interface).
  - Sequences of key strokes can be effectively bound to invoke virtually any existing/user-defined Lisp function.
- It allows sophisticated sequence of key strokes in key bindings, e.g., `Ctrl`+`u` `Ctrl`+`x` `s`.

Agenda

Emacs Key Binding Recommender System (EKBRS)

Lisp function `describe-variable`

function-function module

word-suffix module

posterior-word-suffix | prior-word-suffix

recommends

recommends

recommends

Prefix

Suffix

Ctrl + h   combine   v

Emacs Key Binding Recommender System (EKBRS)

|	prior-word-suffix	posterior-word-suffix	function-function
Exploit relationship between	suffices, English words	suffices, English words	Lisp functions, Lisp functions
Relationship derived from	Prior knowledge	Key binding database	Key binding database
Example	`next-line`/`previous-line` is bound to `Ctrl`+`n`/`Ctrl`+`p`.	Lisp functions with "buffer" in their names are often bound by a key sequence with `b`.	Most help-related Lisp functions are bound by a key sequence with prefix `Ctrl`+`h`.

Agenda

Setup

- We used the default global key binding database in vanilla Emacs 26.1.
- 391 key bindings after removing key bindings involving keys that are outside the main area of a qwerty keyboard.
- Evaluation:
    - Leave-one-out strategy: In each round, we temporarily remove one key binding (for a Lisp function ℓ) from the key binding database.
    - We let EKBRS recommend $n = 1 \ldots 5$ key bindings for ℓ. The key binding is *recovered* if the removed key binding is among the recommended key bindings.
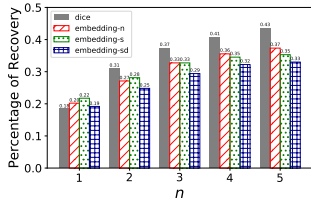   - We use the percentage of recovery as an evaluation metric.
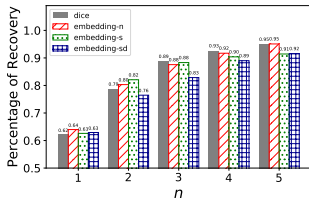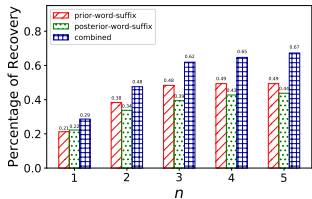
(a) Prefixes (b) Suffices (c) Key bindings

Agenda

Conclusion

- Key bindings for physical keyboards often play critical roles in productivity and user experience. We need recommender systems for key bindings.
- We proposed the first (to the best of our knowledge) recommender system for key bindings, with a demonstration in Emacs.
- Future work: Develop recommender systems for key bindings in other contexts.