2018 42nd IEEE International Conference on Computer Software & Applications

Text Entry on Smartwatches: A Systematic Review of Literature

Mateus M. Luna*, Fabrízzio A. A. de M. N. Soares†, Hugo A. D. Nascimento†,
Joyce Siqueira*, Thamer H. H. Nascimento*, Eduardo F. de Souza† and Ronaldo M. da Costa†
Instituto de Informática, Universidade Federal de Goiás, Goiânia - Goiás - Brazil
Email: {mateus.m.luna, joycitta, thamerthn}@gmail.com*, {fabrizzio, hadn, eduardosouza, ronaldocosta}@inf.ufg.br†

***Abstract*—As an emerging technology that combines mobile and wearable markets, smartwatches are finding their place on consumers' daily lives. They allow tasks that used to be performed only by smartphones and tracking devices. Despite the increasing interest on them, a task that is still not fully covered by these devices is text entry, mainly due to their reduced screen size. Researchers have been working hard on solutions for this issue in the past years, and a number of methods for interactive text entry with smartphones now exist. The aim of this paper is to present a systematic review on these methods, showing what has been developed and what is the performance of the current state of art of the technology. The review focused on four databases. After applying a large selection criterion, it resulted in twenty-six approaches, which helped to answer questions that grounded this work. We hope to deliver a rich and useful foundation about methods, results, challenges and opportunities and to support new research on smartwatches.**

***Index Terms*—Text Entry, Smartwatches, Text Input, Wrist-worn, Systematic Review.**

I. INTRODUCTION

In the last years, smartwatches are starting to take a place in consumer's daily lives. These devices can be used either as an extension of a smartphone or as a standalone device/platform for applications, and are carried persistently by users [1].

Even though a variety of applications are being developed to this novel platform, some interactions are still limited by its physical form. In particular, their small screen size makes them challenging for text entry, resulting sometimes on devices more suitable for viewing rather than inputting content [1]. A problem commonly referred as a "fat finger" is mentioned by researchers for challenging the use of on-screen QWERTY keyboards, as it can be hard to correctly target small buttons.

Many well known smartwatches provide voice input as the preferred input method. Yet, studies show that accuracy is still an issue for such a technology, notwithstanding a lot of improvements that have been made to them [2], [3]. Privacy issues and the embarrassment of "talking to a wrist device" are also mentioned by some users [4]. On the other hand, developing better methods and technologies for text entrance with smartwatches may help to improve even more the overall interaction capabilities of these devices.

The aim of this paper is to present a Systematic Review of Literature (SRL) covering the main published solutions for text entry on smartwatches. The study flags a starting point for future research.

The remainder of the paper is structured as follow: Section II describes the planning and the execution of the SRL. Section III presents, as results, a brief synthesis of the studied literature, highlighting the answers for some search questions and the main characteristics of the existing text-entry methods for smartwatches. In section IV, we present our general discussions and considerations about the theme. Lastly, in Section V, we draw our conclusions.

II. SYSTEMATIC REVIEW OF LITERATURE

Systematic Review of Literature is a method to identify, evaluate, and interpret relevant pieces of research for a particular research question, area or phenomena of interest. In order to conduct a SRL, the protocol defined by Kitchenham [5] with the aid of the software tool StArt [6] were used in the current investigation.

A. Planning protocol

The planning was conducted by PixelLab researchers, from the Federal University of Goiás.

1) Research questions:

Q_1. Which are the existing methods for text entry on smartwatches?

Q_2. Which hardware and/or software resources are required?

Q_3. How were the methods evaluated in terms of effectiveness and efficiency?

2) Databases for literature searching: The study was conducted on four well known literature databases with scientific scope – ACM Digital Library (dl.acm.org), IEEEXplore Digital Library (ieeexplore.ieee.org), Science Direct (www.sciencedirect.com) and Scopus (www.scopus.com).

3) Inclusion criteria: I_1. Proposing a new model for text entry method on smartwatches; I_2. Analyzing difficulties in terms of traditional and existing methods; I_3. Describing a comparison between methods; I_4. Including user studies with speed analyses; and I_5. Presenting user studies with errors rate analysis.

4) Exclusion criteria: E_1. Not describing any new text entry method for smartwatches; E_2. Smartwatches are used as a mere input element to other device, or when the focus is not on the interaction between the user and the smartwatch; E_3. The focus is on wearable devices other than wrist-worn or smartwatches; and E_4. The research does not perform any kind of performance or effectiveness evaluation, only makes propositions.

5) Quality criterion: QC_1. The papers describing text entry speed measures in user studies.

6) Data extraction fields: D_1. Type of Method (TM), separating Hybrid Hardware/Software (require extra hardware components) and Software-only solutions; D_2. Input Source (IS), that can be Screen, Sensors, Camera or Microphone as interfaces for the user and the operating system; D_3. Words per Minute (WPM), indicating the method's average speed in tests; D_4. Error Rate (ER), indicating values for

0730-3157/18/$31.00 ©2018 IEEE
DOI 10.1109/COMPSAC.2018.10242

the method's average error rate in tests, when evaluated; D_5. Single Handed (SH), telling whether the method allows the use of only the hand wearing the watch; D_6. QWERTY (QY), indicating whether the layout was QWERTY-based or not; D_7. Auto Complete (AC), when the system also offers auto-completion or suggestions during typing; D_8. Numbers and Special (NS), informing if the method provides solutions for having the insertion of numbers and special characters; and finally, D_9. Single Step per Character (SSC), informing if the method requires no more than one step (a single click) to perform a character inclusion. This measurement is similar to the Keystroke Per Character (KSPC), sometimes found in the literature, although not as precise, as many of the papers did not explicitly mention it.

In addition to these pieces of information, answers to the above-mentioned questions reported, such as the method created and how it was evaluated.

B. Execution

The choice of keywords for building the search string was based on terms commonly found on the literature and on the market regarding the topic. For the SRL execution, a specific keyword string was formulated and used for each database, as described bellow:

- ACM: *acmdlTitle:(+("smartwatch" "smartwatches" "smart watch" "smart watches" "wrist-worn" "wrist worn" "wristband" "wrist band") +("keyboard" "text entry" "text-entry" "text input" "text-input")) OR recordAbstract:(+("smartwatch" "smartwatches" "smart watch" "smart watches" "wrist-worn" "wrist worn" "wristband" "wrist band") +("keyboard" "text entry" "text-entry" "text input" "text-input")).*
- IEEE Xplore: *(smartwatch* OR "smart watch" OR "smart watches " OR "wrist-worn" OR "wrist worn" OR "wrist band" OR "wristband") AND (keyboard OR "text entry" OR "text-entry" OR "text input" OR "text-input")* "Meta-data", in command search.
- Science Direct: *title-abstr-key((smartwatch* OR smart watch OR smart watches OR wrist-worn OR wrist worn OR wrist band OR wrist bands) AND (keyboard OR text entry OR text-entry OR text input OR text-input)).*
- Scopus: *TITLE-ABS-KEY ((smartwatch* OR "smart watch" OR "smart watches" OR "wrist-worn" OR "wrist worn" OR "wrist band" OR "wristband") AND (keyboard OR "text entry" OR "text-entry" OR "text input" OR "text-input")).*

The search was applied in November 2017. Fig. 1 summarizes the amount the papers processed at the selection step of the SRL, with a total of 135 papers (33 accepted and 102 rejected or ignored ones).

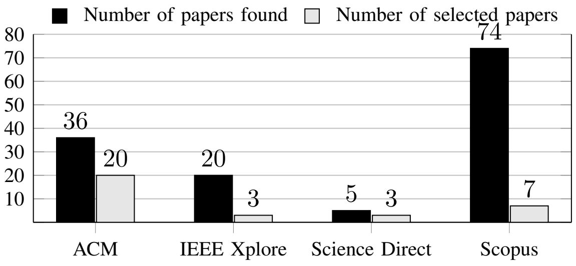

Fig. 1: Synthesis of Data Extraction.

Following, a full-text reading was performed. The quality criterion was verified and the Data Extraction Fields were collected when evaluating each paper. The result of this step is presented next.

III. RESULTS

Some papers were excluded after applying the quality criterion – DragType [38] did not presented any evaluation, and EdgeWrite [39] did not provide any speed test. This ended up with 31 papers, corresponding to 26 text entry methods. An overview of the methods and the extracted data can be found at Table I.

Using the complete information collected, it was possible to answer our research questions, as presented next.

Q_1. *Which are the existing methods for text entry on smartwatches?*

1) ZoomBoard [7]: ZoomBoard uses a scale approach, dividing character insertion in two steps. First, users press on a desired key, in a QUERTY-layout keyboard. Instead of immediate selection, the keyboard zooms in, with a zooming transition, used to preserve perceptual constancy. The central point of this zooming can have different strategies, in this paper a linear combination of zooming to the center and zooming to pressed target was used. Next, with larger targets, the user can refine their finger position if needed, and once again press their desired key. If necessary, more levels of zoom can be employed. Once keys have reached a size that enables accurate section, zooming stops and the key is typed.

2) Funk's method [8]: Funk *et al* proposes a touch-sensitive wristband, to allow free screen space while tapping. Characters are accessed in one side of the wristband, over two different layouts: multitap and linear. The first aggregates three to four letters in one button, where disambiguation is done by consecutive taps. The second provide small targets to each letter, that can be accessed with a finger sliding from top to bottom. Erase and space are the last buttons on both.

3) SwipeBoard [9]: In SwipeBoard's method, keyboard is divided into nine regions, to which the user should perform a swipe: upper-left, up, upper-right, right, lower-right, down, lower-left, left; except for the central region, where a tap anywhere within the interface is used for selection. After the first swipe or tap, the interface will show the selected region of three or four keys, where user should then swipe or click in the direction of the desired key. At this step, the user can also swipe down to return to the full keyboard view. At any time, swiping lower-left twice deletes an entered character, and swiping low-right twice enters a space.

4) Komninos's method [10], [11]: Komninos *et al* proposes a method where watch face is fully covered with three rows of buttons: First one with three letters buttons, aggregated in groups of four to six letters; Second with entered text; Third with remaining of three letters buttons. The method uses an alphabetic layout, and the words distribution is based on a frequency study. A disambiguation algorithm from OpenAdtaptx determines which of the letters user pretended to insert. Spaces are inserted on middle. Swipes allow word completion, alternation between suggestions and special characters.

5) SplitBoard [12]: In SplitBoard, a QWERTY layout is divided into two sections: left and right, displaying six keys per row with a two column overlap. Besides these two main sections, a third section, for numbers and special characters is available. A horizontal flick gesture is used to change

TABLE I: Summary of Extracted Data from Selected Works

Study Method	**Variation**	**TM**	**IS**	**WPM**	**ER**	**SH**	**QY**	**AS**	**NS**	**SSC**
ZoomBoard [7]		Software	Screen	9.30	0.20^{3}	No	Yes	No	No	No
Funk's method [8]	Linear Multitap	Hybrid	Sensors	2.91 3.45	2.58^{3} 3.17^{3}	No	No	No	Yes	Yes No
SwipeBoard [9]		Software	Screen	19.58	4.18%, $13.30\%^{4}$	No	Yes	No	Yes	No
Komninos's method [10], [11]		Software	Screen	8.10	-	No	No	Yes	Yes	Yes
SplitBoard [12]		Software	Screen	14.75	8.54%, $0.58\%^{5}$	No	Yes	No	Yes	No
Leiva's methods [13]	Small (Callout, ZShift) Medium(C,Z) Large(C,Z)	Software	Screen	4.3, 5.4 7.1, 7.2 8.3, 9.1	2.6, 1.3^{3} 0.8, 1.3^{3} 0.7, 0.9^{3}	No	Yes	No	Yes	Yes
VelociTap [14]	Normal Small Tiny	Software	Screen	40.6 38.2 34.9	$3\%^{3}$ $4\%^{3}$ $10.7\%^{3}$	No	Yes	Yes	No	Yes
InclineType [15]		Software	Sensors	6.00	-	No	No	No	Yes	Yes
Virtual Sliding QWERTY [16]		Software	Screen	11.9	-	No	Yes	Yes	No	
InvisiBoard [17]		Software	Screen	9.50	3.2^{6}	No	No	Yes	Yes	Yes
SwipeKey [18]	SwipeKey 4 SwipeKey 5	Software	Screen	11.00^{***} 10.9^{***}	$4.4\%^{3}$ $7.4\%^{3}$	No	No	No	No	Yes
WatchWriter [19]	Tap Gesture	Software	Screen	22.00 24.00	$1.5\%^{3}$ $3.7\%^{3}$	No	Yes	Yes	No	Yes
DriftBoard [20], [21]		Software	Screen	8.77	$1.13\%^{3}$	No	Yes	No	No	No
ForceBoard [22]		Software	Screen	12.4	$9.23\%^{5}$	No	Yes	No	No	Yes
Fujiwara's method [23]		Software	Screen & Microphone	16s*	$76\%^{7}$	Yes	No	No	Yes	No
ETAO Keyboard [24]	Sitting Walking	Software	Screen	12.46 9.36	5.76%, $0.41\%^{5}$ $5.13\%^{5}$	No	No	No	Yes	No
Darbar's method [25]		Hybrid	Sensors	3.9	$6.4\%^{5}$	No	No	No	Yes	Yes
Turner's methods [26]	Trace Tap	Software	Screen	34.11 26.97	0.01^{6} 0.03^{6}	No	No	No	Yes	Yes
DualKey [27]	QWERTY SWEQTY	Hybrid	Screen	19.61 21.59	$5.25\%^{5}$ $3.27\%^{5}$	No	Yes No	No	Yes	Yes
UniWatch [28], [29]		Software	Screen	9.84	-	No	No	Yes	Yes	No
Ilinkin's methods [30]	SKY CJI NRG	Software	Screen	23.4 23.6 22.0	10.5^{5} 12.4^{5} 9.6^{5}	No	No	No	Yes	No
Dunlop's method [31]	Standard Read Focus Write Focus	Software	Screen & Sensors	29.2 27.2 26.7	0.75 - $1.0\%^{3,9}$ 0.75 - $1.0\%^{3,9}$ 1.0 - $1.2\%^{3,9}$	No	Yes	Yes	No	Yes
COMPASS [32]		Software	Bezel	9.30	$<0.25^{3,9}$	No	No	Yes	No	Yes
Nascimento's method [33], [34]		Software	Screen	8.10	-	No	No	Yes	No	No
WatchBoard [35]	Alphabetical QWERTY	Hybrid	Buttons	14.29 15.23	-	No	No Yes	Yes	Yes	Yes
Vouch-T [36], [37]	KeyPad QWERTY	Hybrid	Screen & Microphone	31714.27** 21475.37**	2.86^{8} 2.56^{8}	No	No Yes	No	Yes	Yes

Speed observations: (*) Speed only presented as total time for correcting errors, which average was approximately 16 seconds according to charts shown in [23]. (**) Speed only presented as total number of seconds for typing 20 sentences. (***) Average value on different blocks not provided, thus used only information from last block of test.

Error Rates observations: 1 - Not specified error rate measure and Incorrect and Not Fixed) error rate, respectively. 2 - 100% - Character Recognition Accuracy (%). 3 - Minimum String Distance/Character Error Rate (MSD/CER). 4 - Soft Error (made at first step of insertion) and Hard Error (made at second step of insertion). 5 - Total Error Rate (TER) and (if available) Uncorrected Error Rate (UER), respectively. 6 - Minimum Word Distance/Word Error Rate (MWD/WER). 7 - The percentages of the cases where participants could fix sentences correctly within 40 seconds. 8 - Error only presented as total of delete key pressed in 20 sentences. 9 - Values not presented clearly on paper, thus extracted by estimation on charts.

the section of the keyboard to be displayed. The space and backspace keys are located at the bottom of the screen.

6) Leiva's methods [13]: Leiva *et al* focused on tiny screens, some even smaller than conventional smartwatches. Besides Zoomboard, two other methods were used. Callout shows a tooltip of the current pressed letter, before user release finger to enter it. Similarly, ZShift shows a zoomed view inside a circle of the area where finger is located. In both cases, user can keep sliding finger to refine click. Swipe gestures switch layout for especial characters, insert space and delete letters.

7) VelociTap [14]: VelociTap is a keyboard decoder that uses a probabilistic model to determine key presses and sentence-based encoding to guess user intention. This method proposed and compared different word delimiters, concluding in three strategies: no pace, normal space bar and swipe to the right. One of the studies performed was if the model would allow text entry on small screen sizes, targeting devices like smartwatches. The values tested are: Normal (60mm x 40mm), Small (40mm x 26mm) and Tiny (25mm x 16mm).

8) InclineType [15]: InclineType uses accelerometer data to evaluate watch inclination in character selection. The letters are distributed in a circular, clockwise alphabetic layout, around the screen limits. A tap confirms which letter is selected. This disposition and arrangement was optimized using letter frequency based on Fitts law. Null calibration, Hysteresis and a Jitter are used to provide accuracy on selection.

9) Virtual Sliding QWERTY [16]: Virtual Sliding QWERTY uses an idea similar to Splitboard, but user can swipe in any direction to achieve the desired key in a layout where only some keys are visible on screen. The study analyses different key sizes (3x3 mm, 4x4 mm, 5x5 mm, 6x6 mm, and 7x7 mm) with a fixed spacing between them and different sliding velocity gain when compared to users dragging (1x, 2x, 3x and 4x). Due to the diversity of combinations, reported WPM on Table I is for 5x5mm, with 3x gain.

10) InvisiBoard [17]: The keyboard occupies the whole screen in InvisiBoard, with zero or low opacity to buttons. The layout is inspired in T9 keyboard. User perform swipe gestures and according to the stroke path, a language model is used to determine which should be the word. The recognizer was adapted to consider mostly matches from the beginning or end of the word. A fat swipe allows user to scroll between alternative strings and a swipe left at the bottom delete words.

11) SwipeKey [18]: Shao *et al* makes a series of design studies to develop an input method that would support both tap and swipe on buttons to specify character selection. The final proposed layout has eight buttons, two lines of four buttons, each containing four characters. User performs a swipe inside the button to enter a letter using only one step. Delete and space are realizable with swipes over the text input area.

12) WatchWriter [19]: Standard QWERTY keyboard using a language model for suggesting and auto-complete words. The suggested or/typed button shows the current word and inserts it with a space ahead, which allows the method to not provide a space bar. User can also use gesture strokes (trace) to type complete words at once, reducing KSPC considerably.

13) DriftBoard [20], [21]: DriftBoard contains a movable keyboard QWERTY layout, inside an interactive area, to allow the user to position the desired key over a fixed cursor, while still able to visualize which letter is selected. The confirmation is done automatically by finger up gesture and if no character key is above the cursor at this point, no character is inserted.

14) ForceBoard [22]: ForceBoard uses a QWERTY-like layout, where keys are grouped by two. In a group, if user performs a simple tap, the left character is selected. If user performs a force-tap (press-click with force level above 0.3), the right one is selected. This kind of force-sensitive touch screen is available nowadays in Apple Watch and iPhone.

15) Fujiwara's method [23]: Fujiwara *et al* proposes an optimization on use of Automatic Voice Recognition system for text input in smartwatches. The idea is to correct recognition errors through a custom Phonetic Alphabet input, which was proved to be more effective than spelling letter by letter or using the already consolidated NATO phonetic alphabet. Once speech recognition is complete, user is able to select words through click or swipes and then perform correction, spelling with the custom alphabet.

16) ETAO Keyboard [24]: ETAO favors common letters in English language, giving a one-tap access to 8 characters, and a button to access other 4 areas: first with numbers; two regions with remaining letters and last region with special characters. Delete and space are performed by swipes.

17) Darbar's method [25]: Darbar *et al* places four Hall Effect Sensors over the four corners of a smartwatch. Users need to wear a magnetic device on finger tip or ring, to activate sensor. Using a geometric shape alphabet, combinations of the four sensors activated would be detected as a character. A timeout is initialized once user crosses at least two sensors, to segment consecutive letters input.

18) Turner's method [26]: Turner *et al* uses Swype™ Keyboard application, that allows word-gesture creation, named tracing. The study compares input on tap and trace methods.

19) DualKey [27]: DualKey uses finger identification to allow entering characters on buttons that have two letters associated to them, thus allowing bigger buttons on screen. The study presented two layout: QWERTY and a proposed variation, SWEQTY. For finger identification the prototype uses a combined miniature photo-transistor and optical detector sensor mounted on the index finger.

20) UniWatch [28], [29]: UniWatch is based on previous study of UniGlyph, which aggregates alphabet letters in three buttons, according to the nature of their shape. Disambiguation is performed by a linguistic predictor. Besides the three buttons, layout has a display area, where suggestions and matches are displayed an ranked. Other actions such as erase word, erase character, insert space and access a row with other three buttons to access numbers and special characters, are performed by swipe gestures.

21) Ilinkin' methods [30]: Ilinkin *et al* analyzes three different multitap layouts optimized for Korean alphabet: Sky (SKY), Chon-Ji-In (CJI) and Na-Rat-Gul (NRG) and compare them with a QWERTY-like layout (2SET).

22) Dunlop's method [31]: Dunlop *et al* displays a QWERTY keyboard on watch face, using horizontal swipes to backspace and space. A tilt angle is used to switch between full-screen text display or keyboard display, with a transparency layer and transition during tilt. The angle of 15^o was determined on initial study. Three variances are tested. First shows both text entered and keyboard. Second displays text by default and keyboard when tilted (read focus) and third shows keyboard by default and text when tilted (write focus).

23) COMPASS [32]: COMPASS is focused on allowing non-touch screen interaction on smartwatches (makes possible interactions as wearing gloves). It takes advantage of rotational bezels available in some models in market. Letters are distributed alphabetically in a circular fashion, close to screen border. Three cursors are distributed dynamically over the letters to prevent long rotations. User rotates the cursors with the bezel, and selects it after pressing physical button. A prediction algorithm with language model suggests words in an inner layer of the circular interface. Users can long-press physical button to enter word selection. Using a flick gesture, user can delete last word or character.

24) Nascimento's method [33], [34]: Nascimento *et al* proposes gesture-based text entry using continuous recognition, where suggestions on-screen allow user to enter a character using at most two gestures. While drawing basic strokes, the recognition algorithm already displays the suggestions. If user does not pretend to insert any of the listed results, it can perform a second stroke. After a timeout, the letter is inserted.

25) WatchBoard [35]: WatchBoard is an hybrid hardware and software proposed model, where four to five physical buttons represent aggregated letters and two extra buttons are used for space/selection and backspace. The disambiguation is performed by a statistical model. Four layouts were initially proposed, later narrowed down to the alphabetical one and a variation of QWERTY. Extra typing modes are available trough swipe on smartwatch, where user can use multitap to enter non predicted characters and special ones.

26) Vouch-T [36], [37]: Vouch-T proposes a complementary model, where voice input disambiguate letters pressed on both a 3x4 keypad-like keyboard and a QWERTY keyboard.

Q_2. *Which hardware and/or software resources are required?*

As the *Type of Method* (TM) column exposes, studies elected in this review tend to work with features present in most commercially available smartwatches. The *Input Source* (IS) shows which of these features are usually exploited. This includes mainly a touch screen interaction using clicks, double clicks, long press and swipe gestures. The last one is argued to be interesting for small-size screens, as it requires concerns only about direction, without the need of targeting precision [9], [17]. Sensors, such as accelerometers, also gain attention [15], being a particularity of wearable devices for providing on-wrist interactions.

Fewer cases risk on approaches that demand extra hardware, such as Funk's proposal with sensors on wristbands [8].

Q_3. *How were the methods evaluated in terms of effectiveness and efficiency?*

Most studies opted for a within subject design, when participants face tasks such as typing sentences. The independent variable usually was the text entry method. Test sessions have been distributed over different days to evaluate learning factors, while the amount of participants varies from 5 to 26. After the test execution, participants are usually asked to answer questions regarding their opinion about the method. For the studies more concerned with the impact of the experience, NASA Task Loader Index has been applied [11], [20]. It is notable the use of MacKeinze set of phrases [40] for evaluating text entry.

Effectiveness, or how well the text entry is performed, can be formally measured in literature by the *Minimum String Distance* (MSD), also named Character Error Rate (CER) or Levenshtein distance [41]. Some authors judge these measure limited or misleading when discussing error correction, leading to the use of Total Error Rates (TER), and Uncorrected Error Rates (UER), also mentioned as Not Corrected Error Rate (NER). For some authors, it makes more sense to evaluate effectiveness in terms of word accuracy, thus using Word Error Rate (WER) or Minimum Word Distance (MWD). It was expected to find a clear standard on this measures, but as seen in Table I, different calculations for error rate are used.

Efficiency in text entry methods is frequently measured in terms of *Words Per Minute* (WPM) and so was for all of the selected studies, with the exception of [23] and [37].

Another measurement is the one of *Keystroke Per Character* (KSPC). Funk [8] discuss this constant on his analysis, to compare a linear and a multitap implementation of his method. On DriftBoard comparison with ZoomBoard and SwipeBoard, KSPC is used to express efficiency, as it is affected by the corrections needed after errors [20]. Some authors adopt KSPC as a measurement of error rate, but it works only if the ratio between key pressed and correct character inserted is, by default, equal to one. This value does affect strongly smartwatches input methods when compared to smartphones, because many of the proposed solutions require a disambiguation step for the user to actually insert a character [7], [12], [9]. To these methods it is important to provide a way to return to the first state/step, in case its entry was not performed correctly. In our data extraction, we show in a generic way how this affects each method with the *Single Step per Character* (SSC) column, where the methods marked with "No" require more than one keystroke to be inserted, even without any errors.

IV. DISCUSSION

Overall, the diversity of user study designs indicates a lack of standardization in this field. More mature pieces of work take care of using recognized techniques from the literature, such as the aforementioned MacKeinze set [40]. Even following such standards, it is essential to a Human-Computer Interaction study to provide a comparison with the existing methods. Such comparisons elucidate the impact of different study designs on the results, as happened in the analysis of SwipeBoard [9], that brings an evaluation of ZoomBoard's speed rate achieving 17.08 WPM, almost twice the value reported on its original paper [7]. Later on, in the analysis of DriftBoard [20], ZoomBoard achieved similar values, while SwipeBoards were considerably different. Again, on SwipeKey's evaluation, SwipeBoards performed extremely inferior to what was reported in [18]. Preparing a faithful comparison test can also be a complex task because many studies demand a full-functional prototype implementation.

Two main strategies for overcoming text input challenges on small screens can be found: proposing new layouts for entering characters, usually with more than one interactive action for each symbol, and using language models. The first strategies demands effort for learning and practicing the text entry. The second one can be hard to generalize to many languages, requires extra processing capabilities, and may not deal with specific words such as names and addresses, essential for communication. Comparisons between methods considering the error rate reinforce the idea that faster methods tend to be more error prone, as observed in [12].

The so called fat finger problem had been argued on most of the works as the biggest justification for proposing new layouts. However, some studies, specially the ones in [19], [26], [31], show that, with a strong language model and techniques such as tracing, users can achieve speed and accuracy close to that observed on smartphones, even having a small QWERTY keyboard on screen. This reveals an interesting shift on the studies of some researchers like Dunlop, that begun proposing new layouts in [10], [11] and ended up proposing improvements on traditional QWERTY keyboards [31]. This is reinforced by the fact that most recent Android Wear operating system offer a keyboard similar to the ones in those studies.

A strategy that seems to be employed by many methods is to replace auxiliary buttons such as "delete", "insert space bar" and "switch to numeric keyboard" by swipe gestures. This is positive for reducing the amount of on-screen buttons.

We also note that the scope of text entry studies in smartwatches can go further. More studies should evaluate new methods in situations that are common to wearable devices, such as walking in open spaces, something that has been suggested for future investigation by some authors. Another area to be explored is accessibility. We previously analyzed Braille text entry methods for smartphones [42], [43] and here foresee another open area for study, targeting blind users.

Moreover, we highlight that, as a systematic review of literature, to cover existing techniques on market is not feasible, since most of them are not published in papers. However, during an introductory search, some interesting applications were found and may deserve further attention by researchers. Thus, although out of the SRL protocol, we mention some keyboards which are already being offered to users: Minuum™, Flesky™, and Swype™.

V. CONCLUSION

This comprehensive Systematic Review of Literature demonstrated a wide range of solutions for text entry on smartwatches. From a total of 135 papers found in scientific databases, 33 were selected, resulting in 26 methods that helped answering our questions. Finally, although results of individual studies are generally positive and clearly highlight the substantial potential of text entry on smartwatches, they still present limitations and the problem remains open. As smartwatches are a developing technology, text entry on them presents itself as a huge opportunity for future research.

REFERENCES

[1] A. Visuri, Z. Sarsenbayeva, N. van Berkel, J. Goncalves, R. Rawassizadeh, V. Kostakos, and D. Ferreira, “Quantifying Sources and Types of Smartwatch Usage Sessions,” in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’17. New York, NY, USA: ACM, 2017, pp. 3569–3581, 00000.

[2] J. Timmermann, W. Heuten, and S. Boll, “Input methods for the Borg-RPE-scale on smartwatches,” in *2015 9th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, May 2015, pp. 80–83.

[3] S. Pizza, B. Brown, D. McMillan, and A. Lampinen, “Smartwatch in Vivo,” in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’16. New York, NY, USA: ACM, 2016, pp. 5456–5469, 00000.

[4] S. Schirra and F. R. Bentley, “"It’s Kind of Like an Extra Screen for My Phone": Understanding Everyday Uses of Consumer Smart Watches,” in *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA ’15. New York, NY, USA: ACM, 2015, pp. 2151–2156, 00000.

[5] B. Kitchenham, “Procedures for performing systematic reviews,” *Keele, UK, Keele University*, vol. 33, no. 2004, pp. 1–26, 2004.

[6] S. Fabbri, C. Silva, E. Hernandes, F. Octaviano, A. Di Thommazo, and A. Belgamo, “Improvements in the start tool to better support the systematic review process,” in *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE ’16. New York, NY, USA: ACM, 2016, pp. 21:1–21:5.

[7] S. Oney, C. Harrison, A. Ogan, and J. Wiese, “ZoomBoard: a diminutive qwerty soft keyboard using iterative zooming for ultra-small devices.” ACM Press, 2013, p. 2799.

[8] M. Funk, A. Sahami, N. Henze, and A. Schmidt, “Using a touch-sensitive wristband for text entry on smart watches.” ACM Press, 2014, pp. 2305–2310.

[9] X. A. Chen, T. Grossman, and G. Fitzmaurice, “Swipeboard: a text entry technique for ultra-small interfaces that supports novice to expert transitions.” ACM Press, 2014, pp. 615–620.

[10] M. D. Dunlop, A. Komninos, and N. Durga, “Towards high quality text entry on smartwatches.” ACM Press, 2014, pp. 2365–2370.

[11] A. Komninos and M. Dunlop, “Text Input on a Smart Watch,” *IEEE Pervasive Computing*, vol. 13, no. 4, pp. 50–58, Oct. 2014.

[12] J. Hong, S. Heo, P. Isokoski, and G. Lee, “SplitBoard: A Simple Split Soft Keyboard for Wristwatch-sized Touch Screens.” ACM Press, 2015, pp. 1233–1236.

[13] L. A. Leiva, A. Sahami, A. Catala, N. Henze, and A. Schmidt, “Text Entry on Tiny QWERTY Soft Keyboards.” ACM Press, 2015, pp. 669–678.

[14] K. Vertanen, H. Memmi, J. Emge, S. Reyal, and P. O. Kristensson, “VelociTap: Investigating Fast Mobile Text Entry using Sentence-Based Decoding of Touchscreen Keyboard Input.” ACM Press, 2015, pp. 659–668.

[15] T. Götzelmann and P.-P. Vázquez, “InclineType: An Accelerometer-based Typing Approach for Smartwatches.” ACM Press, 2015, pp. 1–4.

[16] J.-M. Cha, E. Choi, and J. Lim, “Virtual Sliding QWERTY: A new text entry method for smartwatches using Tap-N-Drag,” *Applied Ergonomics*, vol. 51, pp. 263–272, Nov. 2015.

[17] A. Mottelson, C. Larsen, M. Lyderik, P. Strohmeier, and J. Knibbe, “Invisiboard: maximizing display and input space with a full screen text entry method for smartwatches.” ACM Press, 2016, pp. 53–59.

[18] Y.-F. Shao, M. Chang-Ogimoto, R. Pointner, Y.-C. Lin, C.-T. Wu, and M. Chen, “SwipeKey: a swipe-based keyboard design for smartwatches.” ACM Press, 2016, pp. 60–71.

[19] M. Gordon, T. Ouyang, and S. Zhai, “WatchWriter: Tap and Gesture Typing on a Smartwatch Miniature Keyboard with Statistical Decoding.” ACM Press, 2016, pp. 3817–3821.

[20] T. Shibata, D. Afergan, D. Kong, B. F. Yuksel, I. S. MacKenzie, and R. J. Jacob, “DriftBoard: A Panning-Based Text Entry Technique for Ultra-Small Touchscreens.” ACM Press, 2016, pp. 575–582.

[21] T. Shibata, D. Afergan, D. Kong, B. F. Yuksel, S. MacKenzie, and R. J. Jacob, “Text Entry for Ultra-Small Touchscreens Using a Fixed Cursor and Movable Keyboard.” ACM Press, 2016, pp. 3770–3773.

[22] M.-C. Hsiu, D.-Y. Huang, C. A. Chen, Y.-C. Lin, Y.-p. Hung, D.-N. Yang, and M. Chen, “ForceBoard: using force as input technique on size-limited soft keyboard.” ACM Press, 2016, pp. 599–604.

[23] K. Fujiwara, “Error Correction of Speech Recognition by Custom Phonetic Alphabet Input for Ultra-Small Devices.” ACM Press, 2016, pp. 104–109.

[24] R. Darbar, P. Dash, and D. Samanta, “ETAO Keyboard: Text Input Technique on Smartwatches,” *Procedia Computer Science*, vol. 84, pp. 137–141, 2016.

[25] R. Darbar, P. K. Sen, P. Dash, and D. Samanta, “Using Hall Effect Sensors for 3d Space Text Entry on Smartwatches,” *Procedia Computer Science*, vol. 84, pp. 79–85, 2016.

[26] C. J. Turner, B. S. Chaparro, and J. He, “Text Input on a Smartwatch QWERTY Keyboard: Tap vs. Trace,” *International Journal of Human–Computer Interaction*, pp. 1–8, Aug. 2016.

[27] A. Gupta and R. Balakrishnan, “DualKey: Miniature Screen Text Entry via Finger Identification.” ACM Press, 2016, pp. 59–70.

[28] F. Poirier and M. Belatar, “UniWatch - Some Approaches Derived from UniGlyph to Allow Text Input on Tiny Devices Such as Connected Watches,” in *Human-Computer Interaction: Interaction Technologies*, M. Kurosu, Ed. Cham: Springer International Publishing, 2015, vol. 9170, pp. 554–562, dOI: 10.1007/978-3-319-20916-6_51.

[29] ——, “UniWatch: A Soft Keyboard for Text Entry on SmartWatches Using 3 Keys,” in *Human-Computer Interaction. Interaction Platforms and Techniques*, M. Kurosu, Ed. Cham: Springer International Publishing, 2016, vol. 9732, pp. 341–349, dOI: 10.1007/978-3-319-39516-6_33.

[30] I. Ilinkin and S. Kim, “Evaluation of Korean Text Entry Methods for Smartwatches.” ACM Press, 2017, pp. 722–726.

[31] M. D. Dunlop, M. Roper, and G. Imperatore, “Text entry tap accuracy and exploration of tilt controlled layered interaction on Smartwatches.” ACM Press, 2017, pp. 1–11.

[32] X. Yi, C. Yu, W. Xu, X. Bi, and Y. Shi, “COMPASS: Rotational Keyboard on Non-Touch Smartwatches.” ACM Press, 2017, pp. 705–715.

[33] T. H. Nascimento, F. A. A. M. N. Soares, C. B. R. Ferreira, L. L. G. Oliveira, A. S. Soares, P. P. Irani, and M. A. Vieria, “Text input in Smartwatches Based Gestures Using Geometric Shape.” ACM Press, 2016, pp. 1–4.

[34] T. H. Nascimento, F. A. A. M. N. Soares, P. P. Irani, L. L. G. d. Oliveira, and A. d. S. Soares, “Method for Text Entry in Smartwatches Using Continuous Gesture Recognition.” IEEE, Jul. 2017, pp. 549–554.

[35] E. Lockerman, S. Wu, A. Rao, J. Lin, N. Bantoc, and B. A. Myers, “Text entry using five to seven physical keys.” IEEE, Oct. 2017, pp. 291–295.

[36] J. Lee, C. Lee, and G. J. Kim, “Vouch: multimodal touch-and-voice input for smart watches under difficult operating conditions,” *Journal on Multimodal User Interfaces*, vol. 11, no. 3, pp. 289–299, Sep. 2017.

[37] M. Lee and G. J. Kim, “Vouch-T: Multimodal Text Input for Mobile Devices Using Voice and Touch,” in *Human-Computer Interaction. Interaction Contexts*, M. Kurosu, Ed. Cham: Springer International Publishing, 2017, vol. 10272, pp. 208–224, dOI: 10.1007/978-3-319-58077-7_17.

[38] H. Cho, M. Kim, and K. Seo, “A text entry technique for wrist-worn watches with tiny touchscreens.” ACM Press, 2014, pp. 79–80.

[39] K. Ueno, K. Go, and Y. Kinoshita, “Design and Evaluation of EdgeWrite Alphabets for Round Face Smartwatches.” ACM Press, 2016, pp. 177–178.

[40] I. S. MacKenzie and R. W. Soukoreff, “Phrase sets for evaluating text entry techniques,” in *CHI ’03 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA ’03. New York, NY, USA: ACM, 2003, pp. 754–755.

[41] R. W. Soukoreff and I. S. MacKenzie, “Metrics for text entry research: An evaluation of msd and kspc, and a new unified error metric,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’03. New York, NY, USA: ACM, 2003, pp. 113–120.

[42] J. Siqueira, F. A. A. M. N. Soares, D. J. Ferreira, C. R. G. Silva, L. O. Berretta, C. B. R. Ferreira, I. M. Félix, A. d. S. Soares, R. M. d. Costa, and M. M. Luna, “Braille text entry on smartphones: A systematic review of the literature,” in *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, June 2016, pp. 521–526.

[43] J. Siqueira, F. A. A. M. N. Soares, C. R. G. Silva, L. O. Berretta, C. B. R. Ferreira, I. M. Félix, and M. M. Luna, “Braille Ècran: A braille approach to text entry on smartphones,” in *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, June 2016, pp. 608–609.