Dealing with Input Mode Confusion During Dual-Language Keyboard Use

Paraskevas Petsanas, Christos Sintoris, Nikolaos Avouris

Dealing with input mode confusion during dual-language keyboard use

Paraskevas Petsanas[1], Christos Sintoris[1][0000−0001−5350−9108], and Nikolaos Avouris[1][0000−0001−5408−1844]

Interactive Technologies Laboratory, University of Patras, 26500 Patras, Greece
ppetsanas@gmail.com, {sintoris,avouris}@upatras.gr
https://hci.ece.upatras.gr

Abstract. Dual-language or multiple-language keyboards are widely used in many parts of the world, where users need to type in more than one scripts. During use of such keyboards, 'input mode errors' often occur when the user types in one language mode while the keyboard layout is set to a different one. These errors, along with frequent switching between input modes, are a source of confusion and distraction. In this paper we report on a study of this condition among Greek-speaking dual language keyboard users and describe a tool for automatically detecting this kind of errors and dealing with them in the case of an English-Greek keyboard. We also investigate the effects of this input mode error correction tool on a group of typical users during a study that lasted several days.

Keywords: Text input · Input mode · Keyboard interaction.

1 Introduction

Multilingual text entry is prevalent among users who write in non-roman alphabets, such as the alphabets used by Greek [1], Arabic [4, 2], Cyrillic [8], and other writing systems, such as those used in Asian countries [3, 7, 14]. When using the keyboard for text entry, these users frequently alternate between the available language modes by using a key stroke sequence or by using the mouse to select the input mode. A review of relevant literature shows that this is an acknowledged problem among users of multiple keyboard layouts [5, 3, 7, 14], although we are not aware of any studies that report on the extend of this condition.

Input mode errors might occur when users perform tasks that require frequent mode switching, for example when writing non-English texts that require English terms, or when working on multiple concurrent tasks that require application-switching, for example typing an answer in an email composer while the primary task is programming in a code editor. Input mode errors have an adverse effect on user experience [14] since, in order to correct an error, the user has to become aware of it, erase the input, switch input mode and finally retype.

This paper focuses on a study of the occurrence of input mode errors among Greek speaking users. We propose a method for detecting the errors and subsequently a method for auto-correcting such errors without switching input mode.

2 Related Work

A number of software utilities have been developed to mitigate input mode errors that occur while typing (Keyboard Ninja [11], Key Switcher [10], LangOver [12], PuntoSwitcher [15], LayoutFix [13]). Most of them work properly with English and other languages, mostly Russian, but none of them is optimized for Greek, i.e. all have either problems with character encoding when switching between Greek and English, do not support Greek accented characters, or do not support automatic mode switching, i.e. to correct the input mode error, the user has first to select a text portion and then invoke the tool.

Solutions and mechanisms for alleviating the effects of input mode errors have been explored by researchers. Most recently, Lee et al. [14] proposed three methods for tackling input mode errors, offering, in some cases, on-the-fly correction, while others [3, 5] proposed methods for automatic correction of input mode errors. The auto-replace method presented here resembles most closely the auto-switch method proposed in [14]. The cognitive process related to switching between different scripts and the technology that may cause a breakdown of interaction [18] during typing needs to be further studied in the lab and in the wild, in the case of various linguistic contexts. The study and the prototype presented next aim towards this direction.

3 Approach

A novel key-monitor prototype was designed and made available for the purpose of the reported study to a number of Greek-speaking users. The key-monitor's purpose was to detect and log input mode errors while running in the background and to automatically correct such errors. The key-monitor logs the following events: (a) word typed using input mode L, where L denotes the currently selected input language, (b) input mode switch, (c) change of active window, (d) type I input mode error, i.e. the user typing in the wrong input mode, (e) type II input mode error, i.e. the user becoming aware of the input mode error before finishing typing a word and correcting it, (f) input mode error after input mode switch, and (g) input mode error after window change. In order for these events to be detected and logged, it is necessary for the key-monitor to recognize individually typed tokens (words) and to detect mode errors.

3.1 Token Identification

A 'word typed' event is recognized when at least one typed alphabetical character is followed by a *delimiting* character. Delimiting characters are the space character, the return character, and any punctuation mark or special character such as commas, exclamation marks, parentheses, the @ sign etc. Upon such a sequence, a 'word typed' event is logged, along with the input mode used, i.e. Greek or English input mode. Certain keystrokes cause the key-monitor to stop the recognition process for the current word and start anew, such as when the user presses a keyboard arrow, the tab button and the home/end or pageup/pagedown buttons. Finally, compound keystrokes, such as Ctrl+a, are ignored.

Having identified a newly typed word, the next task of the key-monitor is to classify it in one of the two languages, Greek or English.

3.2 Language Identification

After a token has been identified, the language identification task is launched, to determine whether the user used the correct input mode to type the word. The language identification task aims to classify a recognized token into one of three categories, (a) current input language L, (b) alternative input language $L*$, or (c) inconclusive. Language identification relies on the basic dictionary lookup [9] method. Spell-checking dictionaries containing Greek and English words were constructed using publicly available lists of Greek[1] and English[2] words. The dictionaries are incorporated in the key-monitor apparatus in the form of an embedded database. Alternative methods were considered but discarded either due to insufficient implementation for both languages, as, for example, Greek stemmers have a very low performance, or due to overly complicated setup. On the other hand, spell-checking dictionaries were considered sufficient for the task.

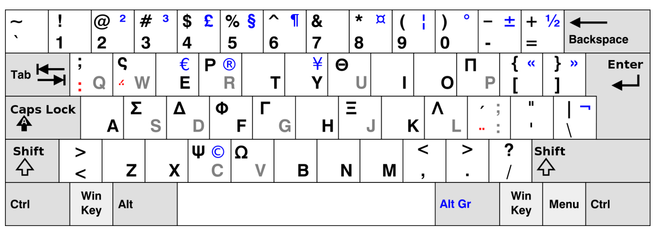

Fig. 1. The standard Greek layout in comparison to the US English one [17]. Greek and Roman letters occupy the same keys. A shortcut alternates between input modes.

The language identification task is aware of (a) the active input mode, (b) the application window currently in focus, and (c) the three most recent language identifications. Otherwise, no knowledge of the user-level context or tasks is assumed, i.e. the key-monitor is not aware if the user casually chats or browses, is involved in a programing task or composes a formal document. The word identified by the token identification task is first looked up in the L dictionary, i.e. the one corresponding to the current input mode. It is then looked up in the alternative $L*$ dictionary. For the alternative dictionary it is necessary to transliterate the word, since English and Greek are written using different alphabets. Transliteration follows the character mapping of the standard keyboard

1 https://www.elspell.gr/
2 https://github.com/dwyl/english-words

layouts for the Greek and English keyboard layouts. For example, the English letter 'u' and the Greek letter 'θ' share the same key on a standard keyboard, i.e. transliteration occurs between characters that occupy the same key (Fig. 1). If the word is found in either one of the L or $L*$ dictionaries, or in neither of them, it is classified accordingly. Some words can be found in both dictionaries, since they have the same key profile, such as the English 'to' and Greek 'τo'. For these words, the three most recent identifications are taken into account and the decision is based on a weighted average of the previous three mode identifications of the words that were typed.

Mode Errors When a word is identified as belonging to the alternative, $L*$, mode, an input mode error is registered. Input mode errors are considered to occur when the user intends to write in mode L and the keyboard is set to the alternative mode $L*$. The key-monitor registers two types of mode errors.

– A *type I mode error* occurs when a full word is typed in input mode L and the key-monitor classifies it as belonging in the alternative mode $L*$.
– A *type II mode error* is registered when the user, before finishing typing a full word, notices the wrong input mode and backtracks to correct the error, i.e. before finishing the token in mode L, the user deletes the typed characters and retypes in $L*$ the same or more characters.

3.3 Automatic Replacement

Upon registering a mode error of type I, the key-monitor (Figure 2) automatically replaces the typed word (typed in input mode L) with the transliterated word (in input mode $L*$). The input mode remains the same.

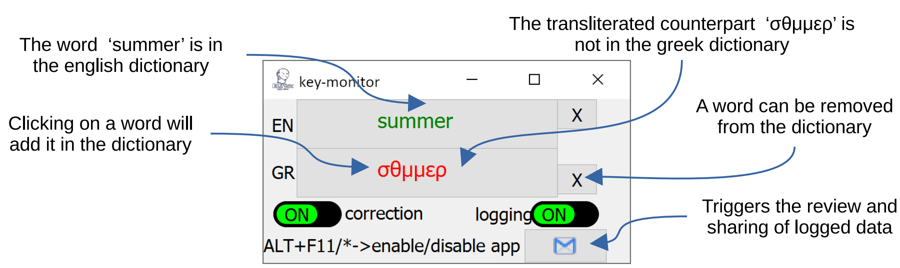

Fig. 2. Overview of the interface of the key-monitor prototype, as was made available to the participants in phase B. The key-monitor runs in the background and registers all keyboard input. In this instance, the user has just typed the word 'summer', which was found in the English dictionary, as indicated by the green color. The transliteration of the word in Greek was not found in the Greek one, as indicated by the red color. The user can add or remove words in the dictionary through the interface.

4 Evaluation study

The purpose of the study was to identify and understand the typing behaviour of dual input mode users, especially regarding the occurrence of mode errors, to evaluate the effectiveness of the auto-replace method and to determine whether it can improve the users' typing experience.

4.1 Participants

To assess the occurrence of input mode errors and evaluate the auto-replace method we conducted a field study involving 24 Greek language users. The study was split in two phases. In phase A, lasting 7 days, keyboard events were silently logged by the key-monitor. In phase B, an updated version of the key-monitor was distributed, which intervened following the auto-replace method described in Section 3.3 in order to assist the users when input mode errors were detected. However, in order to account for any novelty effects, phase B lasted 3 days longer than phase A and we took into account typing data for the last 70% of the typing logs.

Participants were recruited through personal and academic contacts. Most used two modes for input for writing Greek and English text, except for four of the participants, who almost exclusively typed Greek text using the Roman alphabet, a practice known widely as Greeklish [1, 16]. The data related to these four participants was excluded from subsequent analysis, since they intentionally typed using almost exclusively the English input mode. The average age of the remaining 20 participants was 26 years (8 female). Most of the participants were undergraduate students (12), while the rest were office workers (6) and programmers (2). The participants were asked for their typical keyboard use on the computer where they installed the key-monitor. Typical use was classified in one of three categories, namely formal (mostly formal Greek or English text), informal (mostly chat, personal communication), and programming, following the classification proposed in [6]. In particular, based on the participants' answers in the post-study interview, 14 declared that they mostly type informal text, 1 used the keyboard for both formal and informal typing, 2 used it for programming and formal text while 3 were not available for the interview.

In total, all 24 users provided us with logged data for both phases of the study and all but 3 participated in the semi-structured interview after the study.

4.2 Procedure

Informed consent was obtained by all participants. To assure the participants that no private or sensitive data would be exposed, since they would be justified in mistrusting running a key-logging application on their workplace or personal computer, it was stressed that before sharing the collected data, each participant would have the opportunity to review the data and to decline sharing them.

Two versions of the key-monitor prototype were made available to all study participants. In phase A the key-monitor was capable only of detecting and logging input mode errors. The participants were asked to start the key-monitor and leave it running in the background while they conduct their usual activities for 7 days. After reviewing the logged data, every participant chose to submit it for analysis. In phase B, the version of the key-monitor with enabled auto-replace functionality was made available to the participants and they were asked to use it for 10 days (Figure 2). Similarly to phase A, the event logging continued and at the end all participants, after reviewing the logged data, agreed to share it with no exceptions. In both phases the participants could enable and disable logging at will, as they could do with the auto-replace functionality in phase B.

Finally, individual semi-structured interviews were conducted at the end of phase B, in order to gather insights on user experience.

5 Results

Phase A Phase A data show that the participants did encounter problems relating to input mode confusion in their everyday typing. Input mode error rate was computed per participant. On average, the participants made almost 4 times more type I mode errors than they made type II mode errors. In particular, 78.6% of the cases the users didn't notice an input mode error and typed a complete word in the wrong input mode while in the remaining 21.4% of the input mode errors, the users noticed it and backtracked to correct it (type II input mode error). This is in contrast to the results reported by Lee et al. [14] in the evaluation of their auto-switch method for Korean-English users, where they found that in 75.2% of the cases the users noticed and corrected a mode error before completing a word. This difference might be even greater, since the key-logger used in the present study does not take into account words that the user finished typing and then backtracked to rewrite in the alternative layout. The alternative, i.e. that the key-logger is overly sensitive in registering type I mode errors is not plausible, as it is evident in the results of phase B and the participant feedback in the post study interviews. The tools used in the two studies have differences, e.g. in the Korean-English study, input mode prediction is triggered by the space key, which limits the number of times it is triggered and thus it might under-count type I errors. Although differences relating to cultural issues might also explain part of the divergence, nevertheless the observation warrants further study.

The rate of mode errors when the users omit switching to the Greek input mode is the same as when they omit switching to the English input mode, on average 49.5% vs. 50.5%. On average, 10% of the input mode errors occur directly after a mode switch and 9% directly after a window change. These results are again in contrast to [14], where it is reported that in 78.3% of the cases the mode errors occurred immediately after an application switch. Overall it is evident that mode errors pose a non negligible problem in the everyday experience of dual mode Greek-English keyboard users.

Table 1. Mean and standard deviations of the total number of words the participants wrote, the number of mode switches, the number of mode errors per total words and the mode switch rate (Eq. 1). A consequence of the strategy for detecting mode errors is that in phase B, type I input mode errors are note being counted. Since they are automatically detected and corrected, users take advantage of this and continue typing in the wrong input mode.

		Phase A	**Phase B**
Total words	Mean	1695.6	1899.7
	SD	1373.3	1800.7
Total mode switches	Mean	135.2	131.4
	SD	119.9	125.8
Type I mode error rate	Mean	7.37	-
	SD	3.38	-
Type II mode error rate	Mean	26.1	17.3
	SD	23.8	18.6
Mode switch rate	Mean	15.7	19.1
	SD	15.1	11.7

Phase B In phase B, with the auto-replace functionality enabled, the overall typing behavior of the participants was improved while using the key-monitor. To quantify this we use the *mode switch rate* (Eq. 1), which measures the amount of words a user types between two consecutive mode switches,

$$MSR_\phi = \frac{TW_\phi}{MS_\phi} \tag{1}$$

where TW_ϕ is the count of total words written in phase ϕ of the study and MS_ϕ is the count of mode switches in phase ϕ.

On average, the words typed between two consecutive mode switches show a slight increase in phase B, from 15.7 words to 19.1. Twelve of the participants show an increase in MSR_b while two show a high decrease. One of them was not available for post-study interview, while the other reported that during phase A he was writing a long text, a task which rarely required input mode switches.

To further investigate the potential improvement of the auto-replace method, we first identified outliers by calculating the interquartile range (IQR) for the improvement in the mode switch rate, $MSR_{improved}$, where $MSR_{improved} = MSR_B\text{-}MSR_A$, which lead to the exclusion of the data of three participants as outliers. Subsequently, we performed a one-tailed Mann-Whitney test on MSR scores for the two phases, which yielded a statistic W of 267 with p=0.006.

Semi-structured interviews The semi-structured interviews confirm the results of the analysis. Most of the participants found the prototype very useful and stated that it optimized their typing experience, both in dealing with mode errors when they forgot to switch the input mode and in their overall typing efficiency. Two participants mentioned that the auto-replace functionality distracted them from their work, so they occasionally disabled the auto-replace functionality. Both stated that they would prefer the key-monitor to indicate a mode error, more emphatically than it did, without correcting it, e.g. by underlining the corresponding word similarly to spelling correction in word processors. On the other hand, eight of the participants confirmed that they adapted their behaviour in phase B and switched input modes less often. Furthermore, concerning the accuracy of the auto-replace functionality, nine participants stated that they felt that most of the time the key-monitor interfered correctly. Finally, two users mentioned that they used the key-monitor as a 'typing assistant', in order to achieve a better typing experience, i.e. they often omitted switching input modes, a behaviour which lead to their exclusion from the analysis of phase B as outliers, but which is nevertheless indicative of the prototype's usefulness.

6 Discussion

This study focused on occurrence of input mode errors of users of dual language keyboard (Greek-English). It should be observed that the demographic involved in this study (mostly university students and young professionals) during their everyday typing activity quite often have to switch between the two language modes half way through a text, so, as it was evident during phase A of the study, a high number of type I and type II input mode errors were observed. The introduction of the key-monitor prototype has improved both user experience and performance, as the mode switch rate improved significantly after introduction of the developed prototype. Although the results of this study are encouraging, nevertheless a number of limitations need to be taken into account. Firstly, the sample of participants is small and no claims can be made regarding its representativeness, both regarding the participant profiles and the nature of the typing tasks performed in both phases. On the other hand, this study investigates the typing behaviour and the effect of input mode errors of dual-language Greek-English users in their normal conditions of work, without interfering with their everyday activity. Furthermore, the improvements of the auto-replace method are evident both in the quantitative analysis and in the testimonies of the participants themselves. Next steps include improvements in the input mode error detection and the auto-replace method, by investigating the adequacy of alternative methods for language identification [9] and testing, both with a larger sample for a longer period and in controlled settings.

References

1. Androutsopoulos, J.: 'Greeklish': Transliteration Practice and Discourse in the Context of Computer-Mediated Digraphia. In: Standard Languages and Language Standards – Greek, Past and Present. pp. 249–278. Routledge (2016). https://doi.org/10.4324/9781315610580-20.

2. Bianchi, R.: 3arabizi, Greeklish, and SMSki: The hybrid making of language in the age of the internet and mobile technology. Tasmeem. 2015, 1 (2015). https://doi.org/10/gjpht7.
3. Chawannakul, C., Prasitjutrakul, S.: Keyboard layout mismatch error detection and correction system utility. In: The 8th Electrical Engineering/ Electronics, Computer, Telecommunications and Information Technology (ECTI) Association of Thailand - Conference 2011. pp. 488–491 (2011). https://doi.org/10/cwcrcb.
4. Darwish, K.: Arabizi Detection and Conversion to Arabic. arXiv:1306.6755 [cs]. (2013).
5. Ehara, Yo and Tanaka-Ishii, Kumiko: Multilingual Text Entry using Automatic Language Detection. In: Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I (2008). https://www.aclweb.org/anthology/I08-1058
6. Feit, A.M., Nancel, M., John, M., Karrenbauer, A., Weir, D., Oulasvirta, A.: AZERTY amélioré: computational design on a national scale. Commun. ACM. 64, 48–58 (2021). https://doi.org/10/gjp75k.
7. Ikegami, Y., Sakurai, Y., Tsuruta, S.: Modeless Japanese Input Method Using Multiple Character Sequence Features. In: 2012 Eighth International Conference on Signal Image Technology and Internet Based Systems. pp. 613–618 (2012). https://doi.org/10/gjpht2.
8. Ivković, D.: Cyber-Latinica: A comparative analysis of Latinization in Internet Slavic. Language@Internet. 12, (2015).
9. Jauhiainen, T., Lui, M., Zampieri, M., Baldwin, T., Lindén, K.: Automatic Language Identification in Texts: A Survey. Journal of Artificial Intelligence Research. 65, 675 – 782–675 – 782 (2019). https://doi.org/10/gf7b64
10. Key Switcher, http://www.keyswitcher.com/. Last accessed 16 Dec 2020
11. Keyboard Ninja, http://www.keyboard-ninja.com/Download.aspx. Last accessed 16 Dec 2020
12. LangOver, http://langover.com/. Last accessed 16 Dec 2020
13. LayoutFix, https://layoutfix.soft112.com. Last accessed 16 Dec 2020
14. Lee, S., Lee, J., Lee, G.: Diagnosing and Coping with Mode Errors in Korean-English Dual-language Keyboard. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. pp. 1–12. Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10/gf3q8n.
15. PuntoSwitcher, https://yandex.ru/soft/punto. Last accessed 16 Dec 2020
16. Tseliga, T.: "It's All Greeklish to Me!" In: The Multilingual Internet: Language, Culture, and Communication Online. Oxford University Press (2007).
17. Wikimedia, https://commons.wikimedia.org/wiki/File:KB_Greek.svg. Last accessed 16 Dec 2020
18. Winograd, T., Flores, F.: Understanding Computers and Cognition: A New Foundation for Design. Addison-Wesley Longman Publishing Co., Inc., USA (1987).