Design and Evaluation of a Spoken-Feedback Keyboard

André Campos, Pedro Branco, Joaquim A. Jorge
Department of Information Systems and Computer Science
INESC-ID / IST / Universidade Técnica de Lisboa
R. Alves Redol, 9, 1000-029 Lisboa, Portugal
acampos@mega.ist.utl.pt, pjgb@immi.inesc.pt, jorgej@acm.org

Abstract

Speech recognition technologies have come a long way in the past generation. Indeed, they are becoming ever more pervasive in our day-to-day lives, especially in the form of voice-activated menus so prevalent in many automated answering systems. However, speech technologies are still of limited usefulness for large-vocabulary speaker-independent applications in noisy environments, especially where relatively limited computing resources are available as in present-day personal digital assistants (PDAs). Given the popularity of digital cellular phones and text-messaging systems, we describe a generic interface that can be used by any application that need text input by visually-impaired users on this kind of devices. Given the shortcomings of present-day speech recognition technology, we opted to develop three types of keyboards, two predictive, with vocal feedback. This paper, describes the interface development and the usability evaluation results with target users. Our prototype testing scenario included composing short text messages (SMS), and sending them via digital cellular networks (GSM) making it accessible to visually-impaired people.

Keywords

Visually-Impaired users, text-entry, PDA, SMS, predictive keyboards, Short Messaging Service.

1. INTRODUCTION

The recent increases in computing power and autonomy of PDAs, together with continued reductions in price, weight and dimensions, have made these more accessible and widely used. As a consequence, many people organize their lives using these mobile devices. However, for visually-impaired people using such devices is extraordinarily difficult or even impossible, due to the small keyboards and screens characteristic of many of these devices, as the lack of vocal or acoustic feedback on almost all data-entry operations. Our work with students at a Foundation catering to people with special needs[1], revealed that most people suffering from blindness or low vision use such devices mainly for placing or receiving phone calls. Particularly troublesome, if not altogether impossible, is text input with mobile phones to compose and send short SMS messages or text input altogether. This is unfortunate, since many commercial services were introduced, accessed exclusively through SMS.

From our research and market survey we found few existent products that support text input by visually-impaired people. Even the few equipments on the market are too expensive and not sufficiently flexible to be used with other applications. Furthermore users need to know Braille in order to use these devices.

We have studied and developed a special tactile interface with audible feedback provided by a text-to-speech (TTS) system. Our aim was to build a simple and flexible interface to allows visually-impaired users to introduce text in mobile, outdoor settings, in a simple, fast and effective way via a standard commercial off-the-shelf handheld device.

The emergence of new speech technologies makes it possible to use speech synthesis for feedback on input data and application state. The ideal interface would combine this with speech recognition for data and command entry. However, great difficulties speech recognition technologies for mobile devices, still pose many problems, due to resource limitations in handheld computers and poor performance in noisy outdoor environments, with multiple speakers or large vocabularies. Moreover, since speech and reasoning mechanisms use the same brain cognitive resources, it is difficult to dictate text and carry other cognitive functions at the same time [2], such formulating or rearranging ideas. Because of these problems, speech cannot be the only method of input. We studied and developed several touch-typing interfaces using a synthesized programmable keyboard. The difference among the interfaces consisted of the number of keys used for data input, which varied from two to nine. We also developed predictive keyboards, using Hidden Markov Models (HMM) and statistic analysis, to reduce the number of keystrokes required for common text-entry tasks.

We used a user-centred design approach to develop our keyboards. To this end we interleaved prototype designs were with testing and evaluation to iteratively refine the prototype that led to the final system. We used different

[1] Raquel e Martin Saín Foundation in Lisbon, Portugal

methods, including simple techniques such as storyboards, "*Wizard of Oz*" tests and "*quick and dirty*" low-fidelity prototypes for early performance assessment, to more advanced functional prototypes or usability tests accomplished in controlled settings that allowed us to confirm results from theoretical predictions.

The usability tests carried out showed that our interface is easy to learn, especially when compared to Braille keyboards allowing even inexperienced users introduction and sending of text to other mobile devices within less than one hour of training, which represents a distinct progress in relation to the existing technologies.

This remainder of this article is organized as follows. First we describe related work ... In the end we introduced the main conclusions that we reached, and discussed possible developments in the sequence of this project.

2. COMMERCIAL DEVICES

A market survey identified existent systems for text entry. The Mobile DAWN Phone Organizer 5500 is an integrated device for managing personal information in mobile settings, sending email and SMS, to receive and to place phone calls. It uses speech synthesis in English language, as an auxiliary form of output. It is quite bulky (240 x 105 x 45 mm) and expensive. Another system with similar functionality is the The PAC MATE BNS is a similar product which allows access to a Windows system through a PDA, allowing Internet navigation, send and to receive email and manage data with easiness. It uses JAWS [3] technology for Pocket PC. It is also relatively bulky (203 x 152 x 40 mm, weighs 0,7 Kg). and very expensive. A 20 cell Braille portable screen can purchased as add-on for over $1895. We can see that the devices discussed above were developed exclusively for users proficient in Braille, which limits their usefulness to other constituencies. Furthermore all the dimensions of those devices prevent their use in mobile settings. Finally, as these devices were developed for niche markets using specialized hardware, their price is quite high, which prevents widespread use.

3. INTERFACES

With the goal to creating an interface that satisfies the established requirements taken from task analysis and user profiles we developed a five different interfaces, differing mainly on number of keys in which we dived the alphabet for data entry purposes. The simplest approach thus developed is the *bipartite* which splits the alphabet into two keys, such that the first key maps the letters "a" to "m". The letters "n" to "z" are assigned to the other key. Besides these two keys of letters another key serves to enter punctuation plus special characters used in messages, such as space, dot, comma, etc. The remaining keys serve always the same function for the different interfaces.

Just as the name it indicates the *tripartite* interface divides the alphabet into three keys. The chief difference to the previous interface is an individual key to enter digits 0-9.

The *quadripartite* interface has four keys which map to the letter "a-f", "g-l", "m-r" and "s-z" respectively. This interface was the one chosen by users due to the notable balance of its components. Even those that expressed an earlier preference for the *bipartite* keyboard during the selection tests, opted for the quadripartite interface after training for its superior performance. Figure 1 shows the bipartite and quadripartite keyboard layouts as mapped onto the original PDA screen.

Figure 1: Bipartite and Quadripartite keyboards

3.1 Mobile Phone

As an alternative to the interfaces described above, we tried a conventional mobile phone layout. This interface was soon discarded in early tests performed with three users. The main reasons has to do with the size of the keys which were too much small for comfortable manipulation and to the high number of interaction elements (12 keys). Even though the keys seemed to have an acceptable dimension, the Braille characters printed on a tactile screen overlay, would be too close for easy distinction by blind users.

3.2 Dictionary for Text Entry

The traditional techniques of text introduction based on multiple keystrokes to select the intended word letter by letter are slow and tedious increasing the level of effort and frustration unnecessarily to write a given text. To solve this problem we used a dictionary developed done by *Centro de Linguística da Universidade de Lisboa* (CLUL) developed in the late 70's and early 80's through numerous oral interviews to derive a frequency lexicon of the modern spoken Portuguese. The integral version of the lexicon lists the 24000 most often-used words. The version that made available to us contains just the most frequent 8000 entries, which is amply sufficient for everyday use.

3.3 Predictive Keyboards

After the first usability evaluation we selected the quadripartite method as the best interface, and started to develop the functionalities missing from initial prototypes. Additionally we developed two special keyboards using predictive algorithms to speed-up speeding text entry.

This has the same configuration of normal keyboard, but reorders the character list in each key according to previously entered text, using Hidden Markov Models

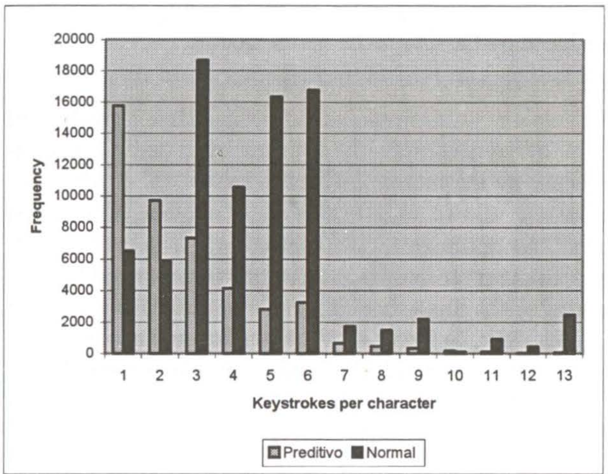

Figure 2: keystrokes per character for normal and predictive keyboards

(HMM). HMMs [10] are finite automata with probabilistic transitions. We use HMMs to determine the most probable letters using information from the last N characters entered. We chose N = 3, since our studies reveled that advantages of larger N were offset both by exponential memory requirements and processing time.

Since the predictive keyboard is based on heuristics, in some situations it may require more keystrokes than a normal keyboard. However, this only happens with very infrequent letters in Portuguese such as "k", "w" and "y", and occasionally with the first letters on each key list. However, the performance gains with common letters such as "d", "e", "o", "p", "q", "r" more than compensate these problems. Figure 2 shows the cumulative keystrokes per character using the *quadripartite* layout for the predictive and normal keyboards for reference corpus extracted from a Portuguese newspaper online edition.

Although the predictive keyboard considerably reduces the number of keystrokes and consequently increase text entry speed, there is an important drawback. The final result of a key press is *apriori* unknown and so the user needs to pay attention to the character disambiguation process in the case the first suggestion isn't the desired one. This is explained by Norman's execution cycle [12]. This couple with the poor performance of the TTS system used, which made it hard to hear some of the characters are responsible for a real performance lower than predicted by the KLM method [7] which we used to determine the optimum performance increase (45% predicted vs 20% revealed by our study). However this is a good result considering that KLM doesn't factor user's fatigue using normal keyboard or general acceptance of the predictive solution.

3.4 Autocomplete

This was the last feature developed, designed on the predictive keyboard by adding word prediction to reduce even more the time and effort, both physical and mental, to enter common words and text. The autocomplete algorithm uses the text already written as prefix to build a list

Figure 3: Example trie

of suggested words ordered by frequency, which contain the prefix. It uses a frequency-annotated lexicon as the source for suggestions. We use a special structure called a *Trie* (derives from Retrieval) [5], to keep string data. This is also known as a prefix tree, to make searching and retrieving by prefix an extremely fast operation. Figure 3 shows a sample *trie*. Our *Trie* was modified to associate each word with its frequency of occurrence in our test corpus. We use it to retrieve only the first N most frequent words with a given prefix. We determined good values for N between three and five. The list shouldn't be too exhaustive so that users do not lose more time browsing the list that would take them to write the word normally.

The algorithm only works with prefixes of least three letters to simplify the process and to spare users from meaningless lists of suggestions. The more frequent word is automatically suggested. Once the user accepts the suggestion the word is automatically completed and a space is added to allow starting the next word. If the word required by the user isn't on the displayed list, this can happen either because the dictionary isn't complete or because that word is not among the top five.

We conducted tests with the same text corpus used to compare normal and predictive keyboards. The autocomplete algorithm automatically completed 1844 out of 3132 eligible words (with more than three letters). Total keystrokes were cut by 18.84%. However, if we factor the 4115 strokes necessary to browse and accept a suggestion the final savings were close to 8.12%. This number can seem lower than the initial expectations but remember that the text was compiled from online newspapers for which our dictionary isn't really adapted,

Figure 4 shows a comparison of the number of strokes necessary to write the text with all three keyboards. According to KLM autocomplete should be **12.24%** faster

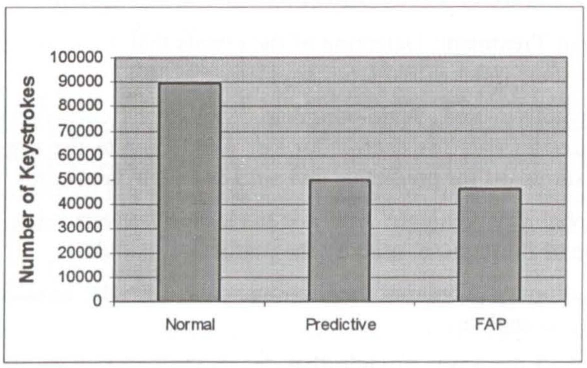

Figure 4: keystrokes for Normal, Predictive and Autocomplete keyboards

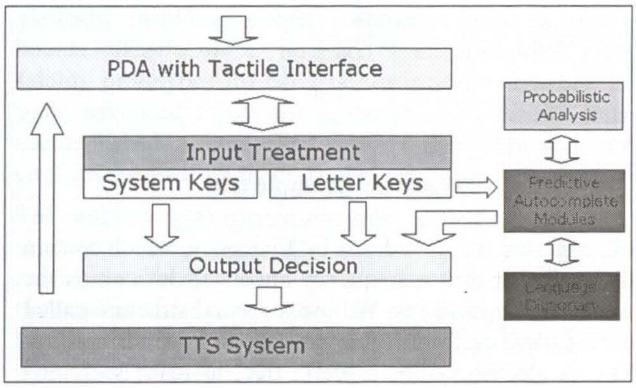

Figure 5: System Architecture

than the predictive and **29.59%** over the normal keyboard. Although on this example, the differences between predictive and autocomplete keyboards are not as significant, due mainly to the necessity of user reorientation when using the PDA's hardware cursor to select or accept a suggestion. This penalty added to slow TTS system response as it synthesizes a complete word, makes the autocomplete keyboard not much faster than the predictive approach. Even though, this was very well accepted by our test users, because of the lesser effort required to perform the same task, thus reducing boredom and frustration.

3.5 Architecture

We present here in a brief way the several modules that compose our system architecture.

PDA with Tactile Interface: It's where the interaction is made with the user.

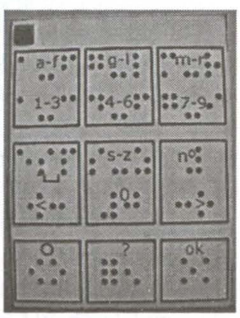

Figure 6: Tactile Interface used in our system

Input Treatment: Detection of the events that occur in the interface (system and letter keys).

Predictive and Auto-complete Module: Reorganizes character lists for each letter key and generate the suggestions list for predictive and auto-complete keyboards.

Probability Analysis: Analysis made to a frequency lexicon of Portuguese, used by the predictive module.

Dictionary: Contains the dictionary used by the module auto-complete.

Output decision: Module that decides which will be the response to the events created by the user.

Speech Synthesis: Entrust of synthesizing the response produced by the Output decision module.

3.6 Operation and functionalities

We select a multi-press method because is relatively well know by the vast majority of users given that its generalized use by the mobile phones industry. The Multi-press technique allows us to reduce the number of required keys, thus optimizing screen space on the device. Whenever the user press a key an event is generated with is treated by the input module, which decides if is a letter or a system key. In the case of a letter key and according on the selected keyboard, the predictive, auto-complete and output decision modules have the responsibility to select the character to send to the speech synthesis module. This decision is made based on the number of times that the key was previously pressed, being automatically increased the index position of the characters list of that key. Based on previously written characters, the list can be ordered alphabetically or dynamically by the predictive module,. When the system timer expires or a different key from the previous is pressed the selected character is added to the final text.

In the case of a system key, and again according to the number of times that it's pressed, a determined function is selected when the timer expires. Examples of functions are: deletion of the last character/word, read or spell the written text or select a suggestion from the list supplied from the auto-complete module.

Many functions are available in our system since we reuse the same key for different functions as we can see in the table below that shows for the number of pressures in the system keys the actions that take place.

4. USABILITY TESTS

We ran two different sets of usability tests. In a first phase we wanted to determine which type of keyboard presented better results, and the second to assess the performance of the final solution and compare it to other approaches.

4.1 Evaluation to the interfaces

The preliminary usability study involved eight users with different degrees of visual impairment and different degrees of experience in digital devices (computer and mobile phone). None of them had ever used a PDA. For each test we started by explaining how the different interfaces work and allowed users to try out each interface. The time required to complete each experiment was on average, an hour and a half. We assessed performance the time required to writing a simple sentence (“`hoje nao posso ir ai, estou doente. ate amanha.`”) for each interface.

The interface type which showed better results at the level of time of execution in the first usability tests, was the quadripartite interface as we can see in Figure 7. The results can be explained by the better division of letters by its keys of the quadripartite interface, making each characters list smaller, and in this way decreasing the number of necessary keystrokes to obtain the wanted

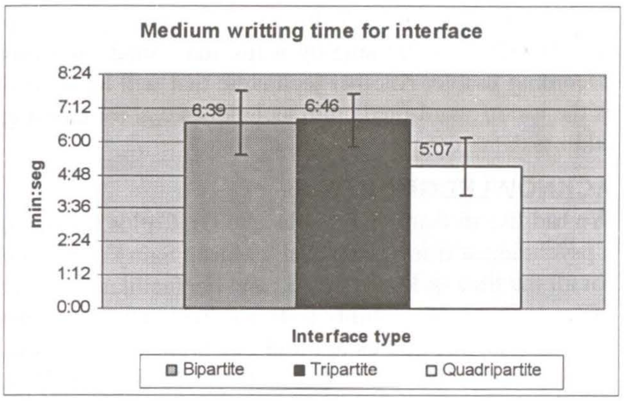

Figure 7: Performance Evaluation

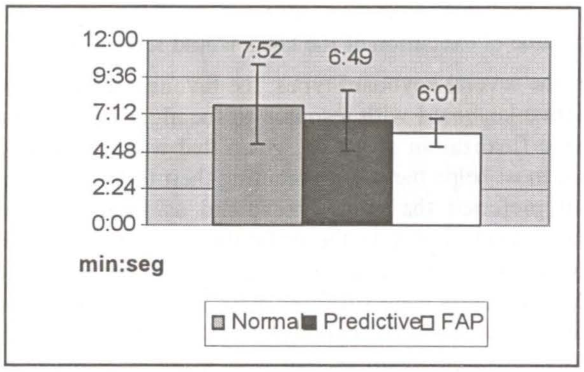

Figure 8: Mean time required to complete test sentence by keyboard type

letter. Looking at the results, we can affirm that the quadripartite keyboard is **23,06%** better than the one split into two interface, with a probability of **83,28%** and **24,38%** better than the tripartite interface, with a confidence of **87,75%** using a t-student bicaudal test. Of course to achieve virtual certainty (confidence over 95%) would require more tests. We went with these values given the early stage of design and that users expressed a clear preference for the quadripartite interface when answering exit questionnaires.

4.2 Evaluation of the different keyboards

Having chosen the interface through the first usability tests, we proceeded to develop all the remaining functionalities and two special keyboards. The first, a predictive keyboard uses HMMs to predict the next character from the last two keystrokes. The second (FAP) adds word-level completion using the dictionary described above. The final usability experiments were performed, using a method similar to the first set by a group of 19 users selected from the target population. Of these, five had already participated in the first usability test and thus were reasonably familiar with the keyboards. Each individual test required accomplishing several tasks:

1. Write the sentence: **"vamos almoçar aos bons dias à 1.20? hoje há cozido à portuguesa."** using each keyboard type (Normal, Predictive and Autocomplete).
2. Insert a contact in the contact list. ("pedro" with the number "934242651")
3. Modify the name of a contact in the contact list ("pedro" to "pedro branco")
4. Send the message "ola" to contact "pedro branco"
5. Delete the contact "pedro branco"
6. Send the message "ola" for a contact not present in the contact list.

Participants were also asked to answer to an informal questionnaire to ascertain their preferences and satisfaction. The chosen sentence allows exercising new functions not implemented in the first prototype, such as accented Portuguese characters. The sentence includes a varied group of characters, letters, numbers and punctuation marks and also alternates between characters in the same and in different keys.

To measure the usability of each keyboard, we again looked at the time required to wrie the sentence. Figure 8 summarizes the experimental results, comparing the predictive and FAP keyboards to the normal keyboard selected earlier (quadripartite interface). We studied the optimal values for each keyboard type for typical sentences using the 8000-word dictionary. The optimal values (assuming error-free operation) predicted less keystrokes for the FAP and predictive keyboards in that order. As we can verify these were experimentally validated by a decrease in the time required to write the sentence, for the predictive and FAP keyboards, respectively.

Using the t-student bicaudal test, the predictive keyboard is **13,35%** faster than the normal keyboard, with **84,90%** confidence. Furthermore, the automatic keyboard was **11,74%** faster than the predictive keyboard, with **89,97%** confidence. While these results fall somewhat short of statistical certainty, the degree of confidence is still high. The difficulties in conducting the usability experiments (each individual experimental took the better part of a day to complete) and the limited resources available to conduct the study, prevented us from conducting a more conclusive study.

To compare our predictive keyboard with a normal mobile phone keyboard with nine keys, we used the Letterwise emulator of Eatoni [6]. The results for the same test sentence used show that even with half of the keys, the predictive keyboard loses only **11,37%**, 114 keystrokes against 104 for the emulator.

4.3 Discussion

The conclusions to extract of the tests are separated in two types: the participants' opinions about the final interface, and the observations on the user's reactions during the execution of the tasks. The great majority of the users considered the interface very pleasant and easy of use, after having one initial phase of adaptation. All of the users said that, with more time of use of the application, the time of execution of the tasks would reduce surely.

Of the several keyboard types, the favourite was the predictive keyboard with resource to the algorithm of automatic finalization of words, given that was the interface that most helps the users executing their tasks. The users that preferred the normal keyboard as first or second choice made it due to the difficulties with the Brazilian speech synthesis engine. Same tends a limited number of keys, the users were surprised with the ratio of functionalities foreseen in the initial phase of the system, and the number of functionalities that were made available in the final prototype.

5. CONCLUSIONS AND FURTHER WORK

With the present work we intended to create a generic interface that allowed the introduction of data in text form in a portable device, in a fast and effective way, for users who suffers from visual deficiencies. In fact, our project allows writing of SMS messages by members of this population segment, in a more accessible way than an of the other solutions available on the market. The results obtained through usability tests are encouraging and they make it possible to conclude that the initial goals were fully met. The general assessment by users was extremely positive considering the results obtained in the usability tests. The whole development was centred on the users, whereby they played an active role in all the phases of its elaboration, continually testing all of the prototypes to find eventual problems or flaws.

It was with the final user in mind that we conceived three versions of the user's manual, one in electronic format to be read in a computer, through a speech synthesizer. Another version of this manual was printed in Braille. Lastly we also prepared a version in audio CD for users that do not have ready access to a conventional computer, or who do not know how to read Braille.

The developed interface is sufficiently flexible to be used in a wide range of products, where it is necessary to insert text, whether its commands or information. In development are already GPS navigation products so that these users can move in places unknown to them without relatives or passers-by help. Other products or possible challenges to address are managing electronic calendars.

However, there are some limitations to the widespread use of our system, namely technological and financial aspects. The required technology is still quite expensive for most users. Even though it is far from the values requested for most electronic devices present in the market for users with visual disabilities. The main cost remains the acquisition of the PDA device, whose value is the most significant part in the total cost, followed by the price of the license of use of the speech synthesis software, indispensable for this work.

We intend to complete the mobile phone interface to allow the reception of email messages, to place and to receive phone calls. We are also looking at synergistic interaction using other modalities such as speech recognition, which is progressing by leaps and bounds, towards becoming usable. Another technique that will be studied is the use of word-prediction techniques e.g. for entering address data.

ACKNOWLEDGMENTS

We had like to thank in first place to Dr. Carlos Bastardo, a psychologist from the Raquel e Martin Saín Foundation for all the time he kindly spared and the facilitation of his office to make the usability tests and the interviews to his students that allowed us to build one prototype with reliability and credibility. We like to thank the National Education department (NOEEE) and especially its staff Cristina Miguel, Maria Idília e João Sequeira for their help in the creation of one tactile interface and for printing the Braille user manuals. Finally for all the users that tested and gave us feed-back about it goes our many thanks.

REFERENCES

1. Gesta at http://www.gesta.org
2. Shneiderman, B., *The limits of speech recognition*, in Communications of the ACM, pp 63–65, Sept 2000.
3. Jaws at http://www.freedomscientific.com
4. Grinter, R. E., and Eldridge, M. A. Y, *Do tngrs luv 2 txt msg?* Proceedings of the European Conference on Computer Supported Cooperative Work - ECSCW (2001). Kluwer Academic Press, Amsterdam 2001
5. Andrew W. Appel and Guy J. Jacobson, *The world's fastest Scrabble program*, Communications of the ACM, 31(5), pp 572–578, May 1988.
6. Eatoni at http://www.eatoni.com
7. I. S. MacKenzie, *KSPC (Keystrokes per Character) as a Characteristic of Text Entry Techniques*, Proceedings of the Fourth International Symposium on Human Computer Interaction for Mobile Systems. 2002
8. MacKenzie, I. S., Kober, H., Smith, D., Jones, T. and Skepner, E., *LetterWise: Prefix-based disambiguation for mobile text input*. Proceedings of ACM UIST 2001. http://www.yorku.ca/mack/uist01.html
9. MacKenzie, I. S.. *Mobile text entry using three keys*. Proceedings of the Second Nordic Conference on Human Computer Interaction., 2002. Also available at http://www.yorku.ca/mack/nordichi2002.pdf
10. O B Sheynin, *A Markov's Work on Probability*, Archive for History of Exact Science 39, 337-377, 1988.
11. Card, Moran, Newell, *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates, Hillsdale, New Jersey 1983.
12. Dix, A. Finlay, J Abowd, G. and Beale, R *Human-Computer Interaction*, 2nd ed. Prentice-Hall Europe 1998.