Towards an Adaptive Communication Aid with Text Input from Ambiguous Keyboards

Karin Harbusch **Michael Kühn**
University Koblenz–Landau, Computer Science Department
Universitätsstr. 1, D–56070 Koblenz, GERMANY
`{harbusch,kuehn}@uni-koblenz.de`

Abstract

Ambiguous keyboards provide efficient typing with low motor demands. In our project[1] concerning the development of a communication aid, we emphasize adaptation with respect to the sensory input. At the same time, we wish to impose individualized language models on the text determination process. UKO–II is an open architecture based on the Emacs text editor with a server/client interface for adaptive language models. Not only the group of motor impaired people but also users of watch–sized devices can profit from this ambiguous typing.

1 Introduction

Written text for communication is of growing importance in e–mails, SMS, newsgroups, web pages — even in synchronous communication situations like chatting, transmitted by electronic devices (computers, cellular phones, handhelds). Computer assisted text entry methods such as ambiguous keyboards are feasible for synchronous and even for asynchronous communication scenarios as they allow complex communication on small electronic devices. Various systems on the mobile phone and handheld market promise a solution to easier and faster text entry.

People with communication disorders are a second group of users who can benefit from computer–assisted text input. Often, speech impairments coincide with severe motor impairments. Standard keyboards or graphical input devices are often unsuitable for motor impaired users. Sometimes, only the operation of one or a very small number of physical switches is possible via buttons, joystick, eye–tracking or otherwise.

These two contexts of use are considerably different: Mobile communication typically happens in the context of asynchronous telecommunication (although fast exchange via SMS or e–mail sometimes develops into a synchronous communication situation). Alternative and augmentative (AAC) methods typically deal with communication strategies in synchronous, face–to–face contexts where, e.g., an electronic communication aid is used to produce a text that is synthesized by a text–to–speech system. (Of course, the produced text can also be utilized in asynchronous telecommunication.)

However, in both contexts the challenging goal is to efficiently produce short pieces of — usually highly variable — natural language text under difficult circumstances. The small size of the device is one factor prohibiting the use of a full keyboard, the other factor is the user's restricted motor function. Both application areas share the aim of a personalized language model to be most effective for the user.

2 Efficient text input methods

Two main classes of efficient text input methods can be identified[2]. First, on a standard QWERTY

---

[1] The project is partially funded by the DFG — German Research Foundation — under grant HA 2716/2–1.

[2] As we concentrate on free text entering devices, we ignore icon–based systems (cf. Lonke *et al.* (1999)).

keyboard, input can be accelerated by *predicting completion* of commands and other word strings (Darragh and Witten, 1992), which reduces the number of keystrokes necessary to enter a word. Motion impaired users who cannot access a full keyboard are slowed down because they have to select each individual key in multiple steps (*scanning*).

Second, *ambiguous keyboards* give rise to communication based on a reduced number of keys (down to 4, cf. Fig. 1). Typing on these devices, the user presses the key corresponding to the letter only once. When the key corresponding to the space bar is pressed, a dictionary is consulted to find all words corresponding to the ambiguous code.

The advantages of an ambiguous keyboard with word disambiguation for users of AAC devices are outlined by Kushler (1998). The efficiency of an ambiguous keyboard approximates one keystroke per letter. Apart from literacy, no memorization of special encodings is required. Attention to the display is required only after the word has been typed. A keyboard with fewer and larger keys may allow easier direct selection for users who otherwise may depend on scanning.

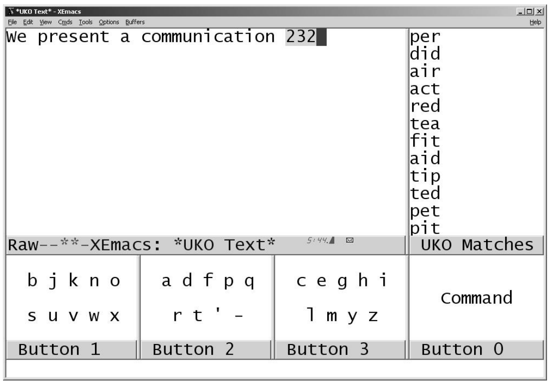
An obstacle to both strategies, prediction and disambiguation, may arise from gaps in the electronic lexicon. If a word is not known to the system, the user of an ambiguous keyboard has to leave the typing mode in order to enter the word by other means. Another drawback of ambiguous text entry is the increased cognitive load imposed on users while typing the word: They may be unable to see the letters of the word already typed and therefore have to memorize the input position.

3 The adaptive UKO-II system

Assistive devices have to respond to dramatically varying needs (Edwards, 1995). Therefore, in order to be useful, they should allow adaptation to specific requirements. We decided basically to design an *open architecture* for a communication system with publicly available sources[3].

Scaffolding for our implementation is provided by the programmable and extendable Emacs text editor, which already includes many text entry and manipulation functions useful in our context. Furthermore, operating system support (e.g. sockets), basic applications like mail, and a development environment with extensive documentation are at the programmer's fingertips. All components in the communication aid dealing with input/output have been implemented as Emacs Lisp modules.

Our communication aid called UKO–II (Fig. 1) is *adaptable* in two ways: First, the system is customizable to differing keyboard layouts and to the selection of word suggestions or additional editing commands. Second, a layered structure of language models controls the disambiguation process and adapts to the user's text input. We discuss both modes in turn.



```
We present a communication 232
```

```
per
did
air
act
red
tea
fit
aid
tip
ted
pet
pit
```

Figure 1: UKO-II Emacs text editing interface with the ambiguous keyboard for English.

Our text entry interface presumes n ($n \geq 1$) physical buttons. This parameter is determined either by the user's motor functions or the device's available buttons. For $n \geq 4$, a genetic algorithm computes a distribution of letters that optimizes the length of suggested word lists with respect to the fixed word frequency information provided by the lexicon. We utilize the frequencies of the CELEX database (Baayen *et al.*, 1995) either for German or English; cf. Kühn and Garbe (2001) for off–line design of the entire keyboard layout. If $n \leq 3$, the keys have to be selected on a virtual keyboard (*scanning*).

In our project the keyboard is tailored to a user with cerebral palsy. No more than four buttons can be accessed directly. Three buttons are ambiguous letter keys with sets of letters assigned to

---

[3] For a collection of Open Source assistive technologies, see TRACE Center (http://www.trace.wisc.edu/linux/).

them. The fourth button invokes letter deletion, command mode or word disambiguation. Words are entered by pressing the corresponding ambiguous key once for each letter. Only after the word is completed, the user disambiguates the input by selecting the intended word in a list of hits provided by the language model. Fig. 1 depicts the situation after the word "aid" has been typed — by pressing the middle, the right and the middle button again (key sequence "232") — and before the user selects the intended word in the list of suggestions[4].

If the target word is not known to the system, it is possible to spell the word and to include it in the lexicon for future use. Other actions in the command mode provide text navigation and editing as well as activation of the speech output system. These actions are triggered either by overloading the three letter keys with commands, or by entering and disambiguating a command name.

The ranking in the list of suggestions for an ambiguous code is determined by a statistical language model. In the simplest case, word frequencies extracted from corpora determine the ordering. As is known from various applications, unconditional probabilities can be improved by adding user–tailored constraints. We provide the user with a *situated* and *personalized* language model consisting of different layers:

1. The *stop word model* comprises a list of a few hundred highly frequent stop words that are not supposed to vary in their distribution with respect to text genres, styles, etc. These words are proposed with highest likelihood if the corresponding code matches.
2. The *local text model* is incrementally constructed while writing a personal document. Recently mentioned words are proposed with higher likelihood than the general model would do (various formulae for shuffling the competitive suggestions are currently evaluated (Harbusch *et al.*, 2003)). Furthermore, we have implemented a word frequency adaptation for the text model.
3. Various *domain specific models* allow appropriate suggestions in different semantic domains such as particular school subjects. Texts in the various domains have been collected. Their frequencies and contextual information are estimated in this layer.
4. The *general language model* stems from large corpora; cf. CELEX frequencies (Baayen *et al.*, 1995). Furthermore, the user can add personal vocabulary such as proper names.

Except for the stop word list, the layers are combined by interpolating the probabilities for any word proposal. Alternatively, the user chooses explicitly between the local text model, a domain model or the general model in order to disambiguate a word.

We have implemented several language models providing the user with ranked lists of predicted words for ambiguous input. Communication between a language model and the text entry interface is handled in a client/server setting implemented by sockets. Sockets enable a clearly distinct interface to the language model components. An interesting technical option of the client/server architecture is to use a *language model server* that is located on another device, e.g. the notebook used in the classroom or the communication aid of another user.

4 Related work

Prediction–based systems are widely applied in the commercial area of communication aids (cf. the PAL system by Swiffin *et al.* (1987) and WordQ by Shein *et al.* (1998)). As we do not investigate prediction–based methods, we only refer to recent work in this area, such as Baroni *et al.* (2002) and Fazly (2002).

An interesting recent development in the area of ambiguous keyboards is the work performed by (Tanaka-Ishii *et al.*, 2002). They describe an ambiguous text input system with five or less letter keys. Word predictions are computed on the basis of *prediction by partial matching (PPM)* at the word level. The letters are assigned to the keys in alphabetical order. This approach favorably compares to ours. However, in our approach the keys have been assigned non–alphabetically after optimisation with respect to a large corpus.

---

[4] In the worst case, this list consists of 50 words in English and 75 in German, respectively.

Other work on typing with word disambiguation focusses on the nine letter keys of a standard phone keyboard (e.g. Forcarda (2001), Skiena and Rau (1996)), and can be traced back to the early 1980s (Witten, 1982, pp. 120–122). Work in alternative and augmentative communication preceding Kushler (1998) deals with key–by–key disambiguation for efficient text input (Levine and Goodenough-Trepagnier, 1990; Arnott and Javed, 1992).

5 Conclusion

We have presented UKO–II, an adaptive ambiguous keyboard providing ranked lists of word suggestions from customized language models.

With respect to the adaptation of the system's user interface, we are transferring the keyboard to a hand–held PC in order to make the every–day use by a wheelchair user more convenient. Providing access to cellular phone communication is also on our agenda.

As to the various language models, we have designed all four layers. On the level of domain models, we have modelled school topics and two different research topics. Currently we run evaluation studies on the competition formulae for the rankings in the final list of suggestions

References

John L. Arnott and Muhammad Y. Javed. 1992. Probabilistic character disambiguation for reduced keyboards using small text samples. *AAC Augmentative and Alternative Communication*, 8(1):215–223.

R. Harald Baayen, Richard Piepenbrock, and Leon Gulikers. 1995. The CELEX lexical database (release 2), [CD-ROM]. Linguistic Data Consortium, Philadelphia, PA.

Marco Baroni, Johannes Matiasek, and Harald Trost. 2002. Wordform- and class-based prediction of the components of German nominal compounds in an AAC system. In *(Tseng, 2002)*, pages 57–63.

CSUN Center of Disabilities. 1998. *Online proceedings of the Technology and Persons with Disabilities Conference 1998*. California State University, Nortridge, CA.

John J. Darragh and Ian H. Witten. 1992. *The Reactive Keyboard*. Cambridge Univ. Press, Cambridge, MA.

Alistair D.N. Edwards, editor. 1995. *Extra-ordinary human-computer interaction: Interfaces for users with disabilities*. Cambridge University Press, Cambridge, MA.

Afsaneh Fazly. 2002. The use of syntax in word completion utilities. MSc thesis, Department of Computer Science, University of Toronto, Canada.

Mikel L. Forcada. 2001. Corpus-based stochastic finite-state predictive text entry for reduced keyboards: Application to Catalan. *Procesamiento del Lenguaje Natural*, 27:65–70.

Karin Harbusch, Saša Hasan, Hajo Hoffmann, Michael Kühn, and Bernhard Schüler. 2003. Domain–specific disambiguation for typing with ambiguous keyboards. In *Proceedings of the EACL workshop on Language Modeling for Text Entry Methods*.

Michael Kühn and Jörn Garbe. 2001. Predictive and highly ambiguous typing for a severely speech and motion impaired user. In C. Stephanidis, editor, *Universal Access in Human-Computer Interaction*, pages 933–937. Lawrence Erlbaum, Mahwah, NJ.

Cliff Kushler. 1998. AAC using a reduced keyboard. In *(CSUN Center of Disabilities, 1998)*.

Stephen H. Levine and Cheryl Goodenough-Trepagnier. 1990. Customised text entry devices for motor-impaired users. *Applied Ergonomics*, 21(1):55–62.

Filip T. Loncke, John Clibbens, Helen H. Arvidson, and Lyle Lloyd. 1999. *Augmentative and Alternative Communication: New Directions in Research and Practice*. Whurr, London, UK.

Fraser Shein, Tom Nantais, Rose Nishiyama, Cynthia Tam, and Paul Marshall. 1998. Word cueing for persons with writing difficulties: WordQ. In *(CSUN Center of Disabilities, 1998)*.

Steven Skiena and Harald Rau. 1996. Dialing for documents: an experiment in information theory. *Journal of Visual Languages and Computing*, 7:79–95.

Andy L. Swiffin, John L. Arnott, and Alan F. Newell. 1987. The use of syntax in a predictive communication aid for the physically handicapped. In Richard D. Steele and William Gerrey, editors, *Proc. of the 10th Annual Conference on Rehabilitation Technology*, pages 124–126, Washington, DC.

Kumiko Tanaka-Ishii, Yusuke Inutsuka, and Masato Takeichi. 2002. Entering text with a four-button device. In *(Tseng, 2002)*, pages 988–994.

Shu-Chuan Tseng, editor. 2002. *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan.

Ian H. Witten. 1982. *Principles of Computer Speech*. Academic Press, London, UK.