EMU - A TEXT PREDICTION SYSTEM TO SPEED UP AND EASE THE TEXT INPUT IN NEARLY ANY APPLICATION FOR DISABLED PEOPLE

G. Seisenbacher*, G. Edelmayer* and W.L. Zagler*

* Institute 'integrated study', Vienna University of Technology, Vienna, Austria

emu@is.tuwien.ac.at

Abstract: A major result of the EC funded project FASTY was the prototype of a text prediction software that showed some very promising results during laboratory and user tests. The software, however, showed also some instability and was not ready for the market in general. Therefore, the decision to develop a system based on the available foundations was made. In this paper information about the background of this decision and a view on the details of the prediction system and the user interface of the EMU software will be given. Some early user feedback will round off the picture.

Introduction

Motor impairments make the use of standard text input devices to the computer difficult and hence slow. But motor impairment often goes together with articulatory deficiencies. Therefore, communication often relies on slow text-input and that leads to communication disorders and negatively influences the quality of life. Whereas experienced typists, for example, will produce some 300 keystrokes per minute, typical mouth-stick users achieve only around 100 keystrokes and input methods using scanning lower this number to around 3 to 10 keystrokes [2].

EMU assists motor, speech, learning and language impaired persons to produce texts faster, with less physical and/or cognitive load and with better spelling. EMU is highly configurable and available for different European languages. It allows easier access to PC-based office systems, to modern forms of IT communication and a faster usage of text to speech synthesizers for voice communication.

Approach

The goal of the EC co-funded R&D project IST-2000-25420 FASTY [1] was the development of software and hardware to ease and speed up the task of writing text with a computer. The project used a two-way approach:

- The text predictor shall make use of both well established as well as innovative new methods [3][7]
- The user interface shall be highly adaptable and specially tailored for the needs of the different users

The final prototype, however, was not ready for the market due to several reasons, some of which – among other things – will be discussed below.

Furthermore, many promising enhancements, which were found during the user tests, could not be included in the final prototype within the project runtime.

Therefore, a new implementation from the ground, based on the knowledge of the project, was developed. The result is the product EMU that is now on the market.

Existing modules

The User Interface module of the FASTY prototype implements several possible ways of displaying predictions to the user. Among these were:

- Free positioning
- Following the caret
- Docked to a certain edge of the screen
- Docked to an application window

The seamless integration and interaction with the operating system, however, is not possible in any way. Furthermore, the code is upgradeable and maintainable very difficultly only. The user interface was sufficient for the laboratory and user tests within the project but not suitable for a product. This part had to be developed from scratch for EMU.

The Language Component of the FASTY prototype uses several methods and algorithms to produce predictions:

- Uni- and bigram-based statistical prediction using general and user frequency dictionaries and a trigram-based Part-of-Speech (PoS) tag model
- Grammar-based prediction analysing predictions in a wide syntactic context and providing syntactic criteria for ranking predictions
- Collocation based prediction for finding word pairs where one word makes the other one “more likely”.
- Compound prediction which allows the creation of compounds “on the fly” without the necessity to store all possible compounds in the dictionaries

Taking the so-called Keystroke Saving Rate (KSR), which is the number of keystrokes saved over the total number of keystrokes, as a basis for the decision which methods to use for EMU shows:

- N-gram based statistical prediction is the common way of almost every text prediction algorithm, it is well approved and shows good results

ISSN: 1727-1983

- Grammar-based prediction increases the KSR by about 0.2 % points at the cost of a rather high system load [4]
- Collocation based prediction increases the KSR by about 0.2% points at the cost of a rather high system load [6]
- Compound prediction increases the KSR by about 1 % point and runs with very little system load [5]

Therefore, EMU uses N-gram based prediction (with PoS tag model) and compound prediction. Figure 1 shows results of laboratory tests with the modified prediction module for several languages. These results promise a good performance for the EMU system.

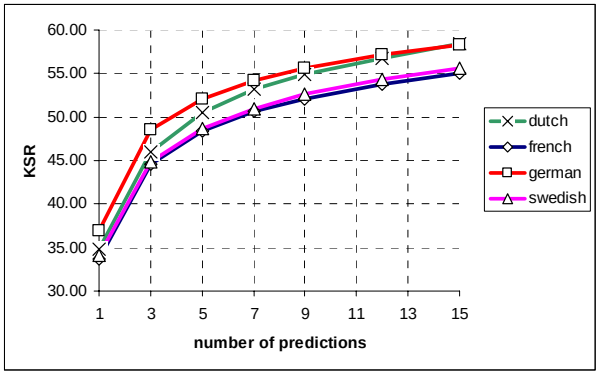

Figure 1: Results of EMU laboratory tests

EMU system

The implemented EMU system uses a rather simple but powerful structure as shown in Figure 2.

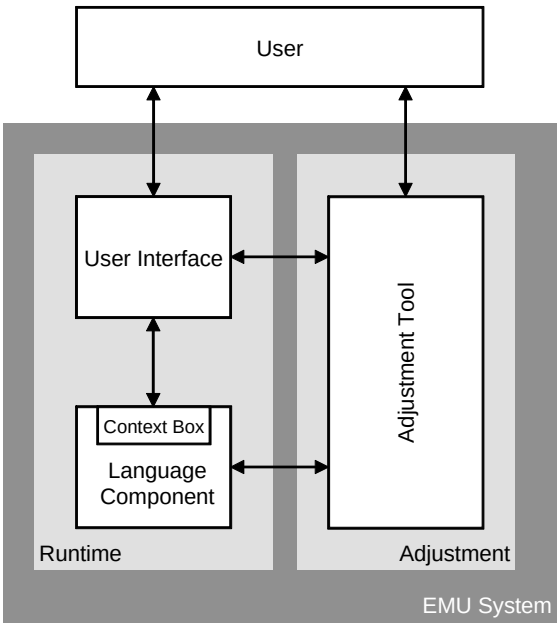

Figure 2: EMU system structure

- Runtime: This program is the actual text prediction system. It monitors the user input and offers words or phrases fitting to the text currently written.
- Adjustment: This program is a very powerful and versatile tool for setting up the Runtime program, maintaining the dictionaries and tailoring EMU to the needs of the user in general.

A more detailed view with the important sub-modules and the connections between them is shown in Figure 3.

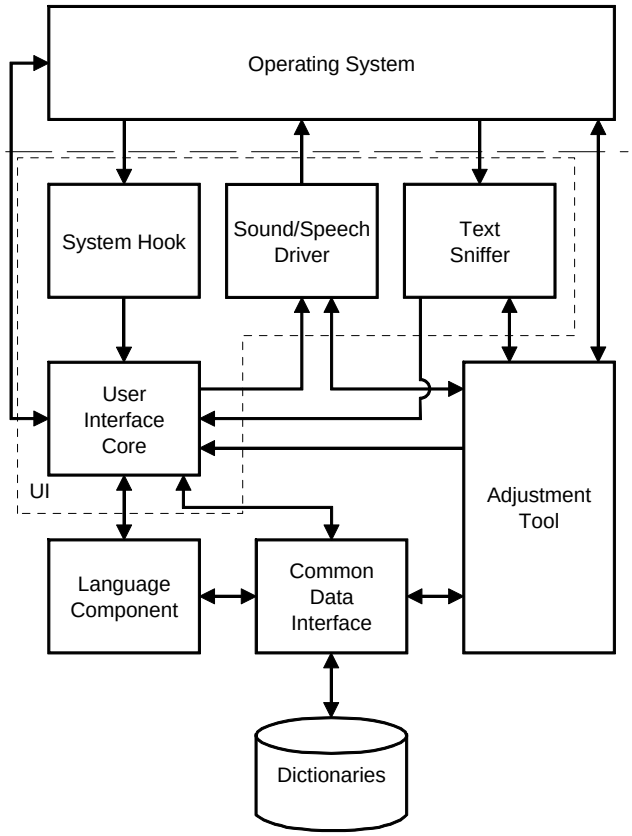

Figure 3: EMU module structure and connections

User Interface (UI)

The User Interface is the active part of the EMU Runtime program. It receives keystrokes, initiates the creation of predictions and presents a certain number of predictions to the user. A screenshot of the currently implemented user interface of the EMU system is shown in Figure 4. The UI may currently be displayed in English, German, Italian and Swedish.

The user interface module is more or less platform dependent, especially the methods of catching user keystrokes and putting a selected prediction back to the system by “pressing” the respective keys from the program is strongly connected to a certain Operating System.

The EMU system is currently available for the Windows Operating System only; porting to other systems means implementing the greater part of the user interface anew.

ISSN: 1727-1983

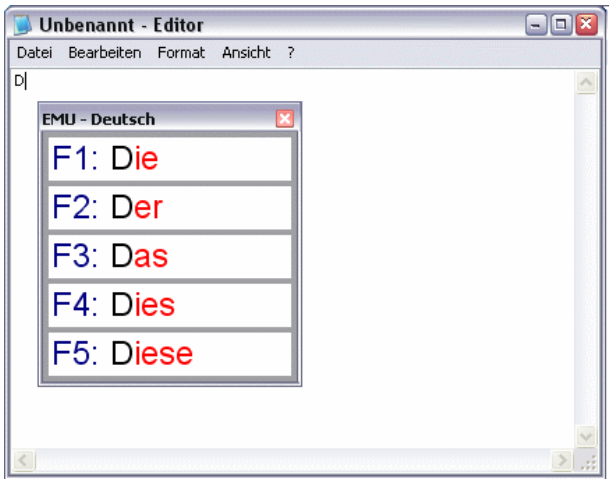

Figure 4: Screenshot of the EMU User Interface

The UI consists of several modules:

- The *User Interface Core* is the central part of the UI. All input events are finally received by the Core; text characters are forwarded to the Language Component (see below). The Core also starts the generation of predictions and presents a list to the user. If the user selects a prediction the Core initiates the "writing".
- Catching user keystrokes is one major task of the *System Hook* module. The module translates different input events to standardised messages which are then sent to the Core. The second major task of the module is inserting characters into the system. This is totally transparent to other applications; all characters seem to be really typed on the keyboard. Finally, the module also monitors the whole system and reports actions like task-switching, changing input focus or starting/stopping applications. Putting these functionalities into a separate module allows the simple replacement with another module using some different method if necessary.
- While output to the screen is done by the UI Core using standard Operating System API calls additional output and feedback by text-to-speech synthesis and different sound events (reading out predictions, indication of the system state through sound output, etc.) is done through a driver system. These *Sound/Speech Drivers* use a standardised interface making it totally transparent to the UI Core.
- One strength of the EMU system is the ability to present fitting predictions right after focusing an input control that already contains text. This ability is implemented by so-called *Text Sniffers*. EMU has a default sniffer module for reading text from standard controls. Additional modules may be linked to the program which allow reading text from applications that do not use standard controls or do not reveal their content in a standard way. The sniffers are also able to find out the read-only state of a control. Again this functionality is offered through a standardised interface.
- Access to system settings and other data is provided by a *Common Data Interface*. This module is also used by the Language Component and the Adjustment Tool (see below) and offers a centralised interface for data access.

Language Component

A block diagram of the EMU Language Component is shown in Figure 5.

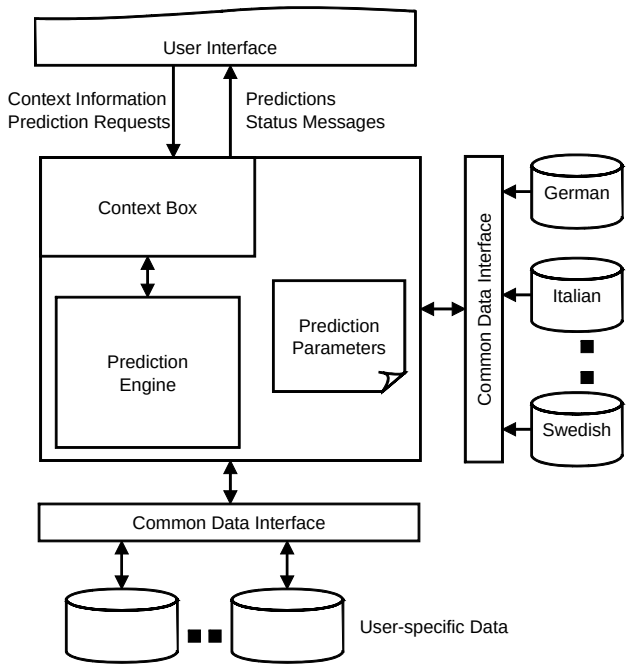

Figure 5: EMU Language Component

The Language Component is implemented platform independent thus allowing to port it to other platforms without (or at least only minimal) modifications.

Furthermore, the module may be switched to another language simply by using another set of dictionaries because the algorithms and methodologies used are language independent [1][7].

The *Context Box* contains the left context of the currently written text. Upon request the *Prediction Engine* generates a list of predictions fitting to the current context. The list of predictions (also containing additional information like probability or PoS information) is passed back to the caller.

The *Prediction Engine* is also able to detect abbreviations in the context and offer the related expansions.

Access to the dictionaries and user-specific data is provided through the *Common Data Interface*. EMU dictionaries are currently available for Dutch, French, German, Italian and Swedish. Most other European languages may be added with limited effort using a set of tools.

ISSN: 1727-1983

The original FASTY dictionaries were based on newspaper corpora; for the needs of EMU they were revised to better fit the needs of day-by-day use.

Adjustment Tool

The Adjustment Tool (AT) runs independently from the Runtime program and is used to setup and configure the latter easily. Figure 6 shows a typical screenshot of the Adjustment Tool.

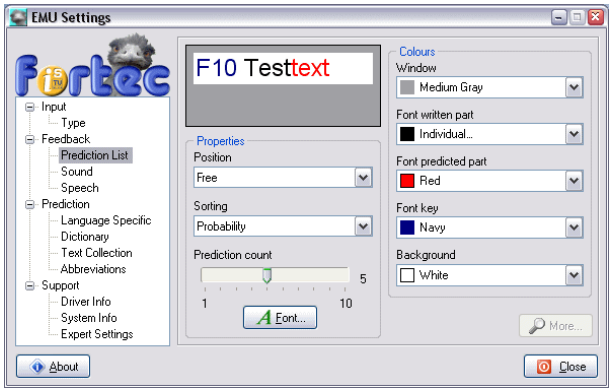

Figure 6: Screenshot of the EMU Adjustment Tool

The Adjustment Tool has access to all dictionaries and user-settings through the *Common Data Interface*. The appearance of the Runtime program may be adjusted to the needs of the user in various ways. Some of these are:

- Position of the Prediction List
- Sorting of the Prediction List
- Colour settings for the Prediction List
- Keys for selecting predictions
- Additional keys for special functions

The Adjustment Tool also has access to the Setup interfaces of the *Sound/Speech Drivers* and the *Text Sniffers* giving the user the possibility to configure all installed modules.

Finally, the Adjustment Tool offers means for dictionary maintenance. Words and phrases may be added, removed or modified. The AT is also able to read text files in many different formats and build dictionaries from the texts. Additionally, maintenance of abbreviation expansion lists is possible (including the import of such lists from different word processors).

User Feedback

EMU has been on the market only for a short period of time now. Therefore, only verbal feedback from some of the users exists.

The tenor of the feedback is that EMU increases the speed of typing as well as the quality of text or, at least, improves the "fun when writing". The users tend to write more and with more details when using EMU.

Conclusion

Judging from the test results and the user feedback EMU seems to be the stable and reliable prediction system aimed for.

A detailed performance and user feedback analysis will be the topic of a follow-up project.

Acknowledgement

EMU is based on knowledge and outcomes of the EC co-funded R&D project IST-2000-25420 FASTY. The project partners were: fortec - Research Group for Rehabilitation Technology, Vienna University of Technology (AT); Österreichisches Forschungsinstitut für Artificial Intelligence (AT); Forschungsinstitut Technologie-Behindertenhilfe der Evangelischen Stiftung Volmarstein (DE); Uppsala University - Department of Linguistics (SE); Multitel ASBL (BE); IGEL Elektronische Kommunikationshilfen GmbH (DE); Seraphisches Liebeswerk für Tirol und Salzburg - Elisabethinum Axams (AT); Ingenieurbüro Jörg-Michael Lindemann (DE); Facultés universitaires Notre-Dame de la Paix (BE).
EMU webpage: http://www.is.tuwien.ac.at/emu

References

[1] Edited Final Report for IST-2000-25420 FASTY Faster Typing for Disabled Persons, http://www.fortec.tuwien.ac.at/fasty

[2] ZAGLER W.L., SEISENBACHER G. (2000): "German Language Predictive Typing – Results from a Feasibility Study", Vollmar R, Wagner R (eds.): Proceedings of the 7th International Conference on Computers Helping People with Special Needs (ICCHP 2000), Karlsruhe, Germany, p. 771-779

[3] MARCO BARONI, JOHANNES MATIASEK, HARALD TROST (2002): "Predicting the Components of German Nominal Compounds", Harmelen F.van (ed.): Proceedings of the 15th European Conference on Artificial Intelligence (ECAI 2002), IOS Press, Amsterdam, p. 470-474

[4] GUSTAVII E., PETTERSSON E. (2003) "A Swedish Grammar for Word Prediction", Master's thesis, Uppsala University, Department of Linguistics

[5] BARONI M., MATIASEK J., TROST H: "Wordform- and class-based prediction of the components of German nominal compounds in an AAC system", Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002), Taipei, Taiwan

[6] BARONI M., MATIASEK J. (2003): "Exploiting long distance collocational relations in predictive typing", Proceedings of the EACL Workshop on Language Modeling for Text Entry Methods (EACL 2003), Budapest, Hungary, p. 1-8

[7] 3rd Edited Annual Report for Publication (IST-2000-25420 FASTY Faster Typing for Disabled Persons), http://www.fortec.tuwien.ac.at/fasty

ISSN: 1727-1983