OpenAdaptxt™:
An Open Source Enabling Technology for High Quality Text Entry

Mark D Dunlop
Computer and Info. Sciences
University of Strathclyde
Richmond Street
Glasgow G1 1XH. Scotland
Mark.Dunlop@cis.strath.ac.uk

Sunil Motaparti
KeyPoint Technologies Pvt. Ltd.
9Q1A Cyber Towers, Hitec City
Hyderabad, India
smotaparti@keypoint-tech.com

Prima Dona
KeyPoint Technologies Pvt. Ltd.
9Q1A Cyber Towers, Hitec City
Hyderabad, India
pdona@keypoint-tech.com

Naveen Durga
KeyPoint Technologies Pvt. Ltd.
9Q1A Cyber Towers, Hitec City
Hyderabad, India
ndurga@keypoint-tech.com

Roberto De Meo
KeyPoint Technologies Ltd
1 Ainslie Rd., Hillington Park
Glasgow G52 4RU. Scotland
rdemeo@keypoint-tech.com

Abstract
Modern text entry systems, especially for touch screen phones and novel devices, rely on complex underlying technologies such as error correction and word suggestion. Furthermore, for global deployment a vast number of languages have to be supported. Together this has raised the entry bar for new text entry techniques, which makes developing and testing a longer process thus stifling innovation. For example, testing a new feedback mechanism in comparison to a stock keyboard now requires the researchers to support at least slip correction and probably word suggestion. This paper introduces OpenAdaptxt: an open source community driven text input platform to enable development of higher quality text input solutions. It is the first commercial-grade open source enabling technology for modern text entry that supports both multiple platforms and dictionary support for over 50 spoken languages.

Author Keywords
Touch-screen keyboard design optimization

ACM Classification Keywords
H.5.2 User Interfaces: Input devices and strategies

Figure 1: Sample interaction with OpenAdaptxt Android Demo – upper panel per image shows text typed with the lower widget showing OpenAdaptxt suggestions

Motivation

User studies and user-based evaluations form the core of much work in text entry. This research involves, for example, considerable work in understanding how people text, to develop new text entry methods for standard input devices, to extend text entry to new technologies and to improve text entry support for users with special needs.

There is a long history of work in text entry research, in trying to understand how people type. Fitts' law [5] modeling has been used extensively to, for example, predict typing times on mobile phones (e.g. [14]) or to optimize new keyboard layouts (e.g. [18]). Different input devices, whether full-sized keyboards, touch screens or miniature keyboards, lead to different typing errors and understanding these is also a key line of research to help improve correction algorithms (e.g. [1, 2]). It is also vitally important that we understand the impact of text entry on other activities such as driving (e.g. [4]).

There have been many approaches to improving text entry for standard devices and to provide efficient input on new technologies. For example the classic T9™ predictive algorithm [6] considerably improved input on classic 12-key phones. More recently, work has been done on, for example, eyes-free text entry on touchscreen phones [15], exploiting multimodal interaction [3], typing on the back of tablets [13] and on tabletop interactive surfaces [7].

Due to their special needs, some users need alternative solutions to text entry. For example visually impaired users are badly affected by touch screen interaction (e.g., a key motivation behind [15]) while fine motor control and eyesight reduction can seriously impact elderly users' performance (e.g. [17]). Other users have more severe motor control limitations and need specialist solutions (e.g. [11]).

Furthermore, with notable exceptions (e.g. [8, 10]), the vast majority of text entry studies have been in English. Kristensson [9] identified localization as the one of the five key challenges for intelligent text entry methods. However, modern text entry methods need detailed dictionaries with language modeling making the cost of researching in new languages high.

Standard text collections ([12, 16]) have helped considerably in providing both reliable and easy to use phrase sets for evaluation, but to develop and evaluate new approaches like those mentioned above almost always involves developing a new text entry engine. As the domain and technologies progresses the base line quality for this engine increases constantly, making it increasingly difficult to develop the systems and thus to innovate.

Overview of OpenAdaptxt

OpenAdaptxt is a new initiative to build an open source community text entry solution. The OpenAdaptxt prediction engine core supports:

- **Error correction**: correction for both traditional spelling errors and mis-taps common on small devices (e.g. missing letters, duplicate letters and hitting adjacent keys) (Fig. 1 top);

- **Intelligent word completion:** using linguistic models to offer meaningful word completions as the user types (Fig. 1 middle);
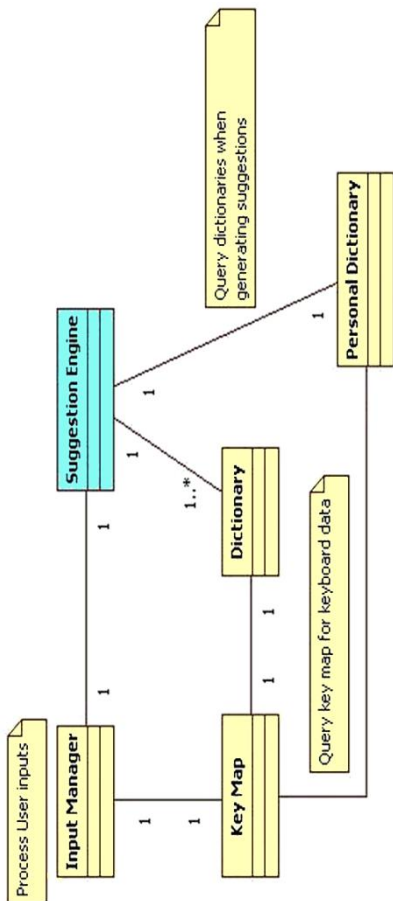
Figure 2: Suggestion engine main components

- **Next word prediction:** based on relationships between words (Fig. 1 bottom);

- **Learning users' individual writing styles** – both user defined words and the pattern of their usage are learned and used to improve predictions (Figure 1 shows interaction after the user has previously typed the workshop name);

- **Over 50 languages**: language dictionaries that support both vocabulary and intelligent prediction are currently available for English (UK & US), French (FR & CA), Italian, Spanish (ES & LA), German, Portuguese (PT & BR), Dutch, Norwegian, Swedish, Finnish, Danish, Polish, Czech, Hungarian, Slovak, Romanian, Greek, Turkish, Russian, Serbian, Croatian, Bulgarian, Slovenian, Lithuanian, Catalan, Latvian, Filipino, Galician, Ukrainian, Estonian, Vietnamese, Indonesian, Icelandic, Malay, Basque, Belarusian, Hebrew, Arabic, Urdu, Persian, Hindi, Marathi, Malayalam, Tamil, Kannada and Telugu.

- **Keyboard flexibility:** programmable support for keyboard layouts (plus standard provisions including Qwerty, Azerty, 12-key phones etc.).

OpenAdaptxt is currently available for multiple platforms such as Android™, Windows™ and Linux™. It has a platform abstraction layer and can be ported easily to new platforms. The OpenSource release includes demo applications for Android in Android-Java (see Figure 1), the OpenAdaptxt prediction core and tools to create new add-ons (including new dictionaries to support new languages and new key-maps to support new keyboard layouts). Figure 2 outlines the structure of the core word suggestion element of the OpenAdaptxt core engine.

Documentation and header files are available from SourceForge[1] with source files available through a SourceForge SVN repository.

Why Open Source?

Researchers, OEMs and application developers have long been suffering from proprietary solutions in the text input space. The leading proprietary solutions have been slow in innovating and are often more interested in protecting patent income than helping end users achieve efficient and effective text entry on a wide range of devices. Acknowledging the major shift towards adoption of open and open source software in connected devices, we believe that an open source de facto standard for text input is the way forward.

By releasing the established Adaptxt text input platform as open source software and launching a transparent community project around its development, the aim is to provide a technically strong, open platform for innovating without the high costs associated with developing individual solutions. We also believe that the collective efforts of the community combined with KeyPoint's on-going contributions to the project will mean OpenAdaptxt itself will progress at a much faster pace and in many new directions than a single group or company can achieve alone.

We believe open source simply creates better software and better products. When everyone collaborates passionately, quality wins! We think the openness also

---

[1] http://sourceforge.net/projects/openadaptxt/

contributes to making software more secure and amenable to integration.

References

[1] Arif, A. and Stuerzlinger, W. Predicting the cost of error correction in character-based text entry technologies. In *Proc. CHI 2010*, ACM Press (2010), 5-14.

[2] Clawson, J., Lyons, K., Rudnick, A., Iannucci, R. A. J. and Starner, T. Automatic whiteout++: correcting mini-QWERTY typing errors using keypress timing. In *Proc. CHI 2008*, ACM Press (2008), 573-582.

[3] Dearman, D., Karlson, A., Meyers, B. and Bederson, B. Multi-modal text entry and selection on a mobile device. In *Proc. Graphics Interface 2010*, Canadian Information Processing Society (2010), 19-26.

[4] Drews, F., Yazdani, H., Godfrey, C., Cooper, J. and Strayer, D. Text Messaging During Simulated Driving. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 51, 5 (2009), 762-770.

[5] Fitts, P. M. The information capacity of the human motor system in controlling the amplitude of movement. *J. Experimental Psychology*, 47, 6 (1954), 381-391.

[6] Grover, D. L., King, M. T. and Kushler, C. A. *Reduced keyboard disambiguating computer* Tegic Communications, Inc., Patent US5818437 (1998).

[7] Hinrichs, U., Hancock, M., Carpendale, S. and Collins, C. Examination of Text-Entry Methods for Tabletop Displays. In *Proc. TableTop 2007*, IEEE Computer Society (2007), 105-112.

[8] Joshi, A., Dalvi, G., Joshi, M., Rashinkar, P. and Sarangdhar, A. Design and evaluation of Devanagari virtual keyboards for touch screen mobile phones. In *Proc. MobileHCI'11*, ACM (2011), 323-332.

[9] Kristensson, P. Five Challenges for Intelligent Text Entry Methods. *AI Magazine*, 30, 4 (2009), 85-94.

[10] Liu, Y. and Räihä, K. Predicting Chinese text entry speeds on mobile phones. In *Proc. CHI2010*, ACM (2010), 2183-2192.

[11] MacKenzie, I. S. and Ashtiani, B. BlinkWrite: efficient text entry using eye blinks. *Univers. Access Inf. Soc.*, 10, 1 (2011), 69-80.

[12] MacKenzie, I. S. and Soukoreff, R. W. Phrase sets for evaluating text entry techniques. In *Proc. CHI 2003*, ACM Press (2003), 754-755.

[13] Scott, J., Izadi, S., Rezai, L., Ruszkowski, D., Bi, X. and Balakrishnan, R. RearType: text entry using keys on the back of a device. In *Proc. MobileHCI 2010*, ACM (2010), 171-180.

[14] Silfverberg, M., MacKenzie, I. S. and Korhonen, P. Predicting text entry speed on mobile phones. In *Proc. CHI'00*, ACM Press (2000).

[15] Tinwala, H. and MacKenzie, S. *Eyes-free text entry with error correction on touchscreen mobile devices*. ACM, City, 2010.

[16] Vertanen, K. and Kristensson, P. O. A versatile dataset for text entry evaluations based on genuine mobile emails. In *Proc. MobileHCI'11*, ACM (2011), 295-298.

[17] Wright, P., Bartram, C., Rogers, N., Emslie, H., Evans, J., Wilson, B. and Belt, S. Text entry on handheld computers by older users. *Ergonomics*, 43, 6 (2000).

[18] Zhai, S., Hunter, M. and Smith, B. Performance Optimization of Virtual Keyboards. *Human-Computer Interaction*, 17, 2 (2002), 229-269.