Maejo Int. J. Sci. Technol. **2011**, *5*(02), 266-278

Maejo International Journal of Science and Technology

ISSN 1905-7873
Available online at www.mijst.mju.ac.th

Full Paper

A one-mode-for-all predictor for text messaging

Vimala Balakrishnan[1,*], Sim Foo Guan[2] and Ram Gopal Raj[1]

[1] Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia

[2] Faculty of Information Science and Technology, Multimedia University, 75450 Ayer Keroh, Melaka, Malaysia

* Corresponding author, e-mail: vim.balakrishnan@gmail.com

Received: 30 September 2010 / Accepted: 8 August 2011 / Published: 9 August 2011

Abstract: This paper discusses the enhancements made on the current mobile phone messaging software, namely the predictive text entry. In addition, the application also has a facility to abbreviate any unabbreviated words that exist in the dictionary, so that the message length can be reduced. The application was tested in a computer-simulated mobile environment and the results of the tests are presented here. These additional features will potentially enable users to send messages at a reduced length and thus reduce the cost of sending messages. Moreover, users who are not adept in using the abbreviations can now do so with features made available on their mobile phones. It is believed that these additional features will also encourage more users to use the predictive software as well as further improve users' messaging satisfaction.

Keywords: text messaging, predictive text entry, abbreviator, mobile phones

---

INTRODUCTION

Short message service (SMS) is a text messaging technology that has become a huge hit among the mobile phone users, especially the youngsters. It is defined as a digital telecommunications protocol that allows the user to exchange short messages (160 characters or less) via mobile phones and other devices such as the computer and Personal Digital Assistants [1].

The mobile phone text messaging phenomenon found its footing during the design of the global system for mobile communication (GSM) technology. In the 1980s, it was found that sending a short message at the same time as speech is possible, and thus the delivery and receipt of non-voice, or specifically, alphanumeric text-based messages, now popularly known as SMS, began. Today, SMS has become the most widely used mobile application worldwide. It has gained

Maejo Int. J. Sci. Technol. **2011**, *5*(02), 266-278

phenomenal success among its users, especially among the youth [2]. The reason for this has a lot to do with convenience, accessibility, as well as cost. SMS is relatively cheaper compared to all the other types of mobile communications such as voice calls and Multimedia Messaging Service (MMS). The average SMS rate in Malaysia is currently only 0.15 MYR (0.05 USD) per SMS while a typical MMS rate rangees between 0.25-0.5 MYR (0.08-0.16 USD) per MMS [3].

Whilst the popularity of SMS continues to grow, the SMS protocol imposes some serious restrictions and raises flaws in usability, as it was not originally designed for mobile phones. An SMS is entered via a phone keypad using the multi-tap or the predictive method. The main limitation of these methods is due to the standard layout of the keypad that most mobile phones have (Figure 1). The keys are overloaded to accommodate alphabets, numbers and also symbols. Therefore, the task of message composition can be cumbersome and time consuming. Other input devices, such as mini QWERTY keyboards and touch screens are on the rise, although the standard 12-button-keypad-based mobile phones remain vastly common in the current market.

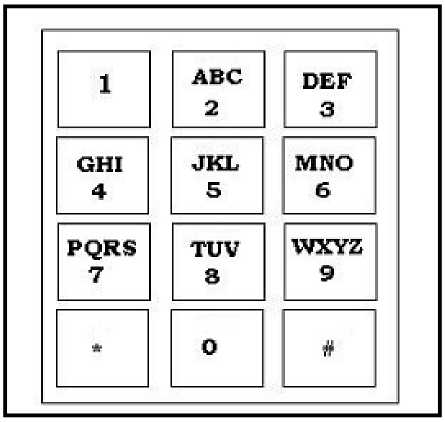


Figure 1. Standard ISO 12-key keypad

SMS TEXT ENTRY

With the limited keys on a standard mobile phone keypad, many vendors have introduced various ways of text input into a mobile phone. However, multi-tap and predictive entry remain the two dominant methods for text entry.

Multi-tap

Multi-tap is implemented on all mobile phones with SMS capabilities. It disambiguates by requiring the user to press a key once or multiple times to enter the desired character(s). For instance, a user needs to press the key-2 once for ‘a’, twice for ‘b’, and thrice for ‘c’. Figure 2 below shows the sequence of key presses needed to enter ‘The cat is seeing.’

The <!> notation is to signify the timeout interval needed to clear the input buffer, since the character to be entered next resides on the same key. This is called segmentation and can be done either by waiting for the timeout mentioned above or by pressing the timeout kill key [4]. The

Maejo Int. J. Sci. Technol. **2011**, *5*(02), 266-278

8	44	33	0	222	<!>	2	8	0	444	7777	0	7777	33	<!>	33	444	66	4
T	H	E		C		A	T		I	S		S	E		E	I	N	G

0 - spacebar

Figure 2. Multi-tap sequence

example clearly shows that a total of 34 key presses are needed to yield only 17 characters (with the space included). Text entry rates for multi-tap are commonly 7-15 words per minute, and with the segmentation issues involving timeout intervals, it is obvious that multi-tap can be slow and tedious [5].

Predictive Entry

Dictionary or lexicon based methods were introduced to increase the speed of text messaging. There are many variants to these methods, the most popular one being T9® by Tegic Communications [6]. The predictive mechanism works by attempting to anticipate the next string of characters entered by a user based on the existing words in a dictionary. Generally, when the first character is entered, the software will offer the most probable words beginning with the particular character(s). Disambiguating words in this manner is not always perfect as the same key sequence may produce two or more words, a phenomenon known as collision. In this scenario, the software will suggest the most probable words and the user may then traverse through the words using a special key (either Next or the asterisk key). The general steps in the predictive method are given below [7]:

i. User enters a code sequence of the desired text; for example, ‘lazy’ is entered as ‘5299.’
ii. The software decodes the code by looking for corresponding targets in the dictionary. The potential words are sorted in a relevant order using linguistic knowledge or according to frequency of most often used words, with the highest frequency word sorted to the top of the list. An example below shows a collision generated by the code sequence ‘5299’:

    Jazz
    Lazy
    Lawy

iii. The user chooses the target word by scrolling with either the asterisk key (e.g. in Nokia) or the up or down navigational key (e.g. in Sony Ericsson).

Figure 3 shows the sequences needed to enter ‘The cat is seeing’ using T9®. A total of only 18 key presses are needed as opposed to 34 key presses on multi-tap, proving that the predictive method is more efficient than the multi-tap. The former usually results in a decreased number of key presses required to write a sentence and hence increases the text entry speed. In addition, the physical effort required to compose a message is also reduced. Unfortunately, this efficiency rate drops very drastically when the user attempts to enter words that are not in the dictionary, be they English words or understandably unlisted non-English words [8]. It is also impossible to enter

Maejo Int. J. Sci. Technol. **2011**, *5*(02), 266-278

8	4	3	0	2	2	8	*	0	4	7	0	7	3	3	4	6	4
T	TH	THE		A	CC	ACT	CAT		I	IS		S	RE	SEE	SEEI	SEEIN	SEEING

0 – spacebar; * – end of selection

Figure 3. Predictive T9® sequence

numerals, acronyms or any combinations of letters and numerals (e.g. 'l8r' for 'later'). Users are then required to multi-tap the desired word into the original message. Manually switching between the text entry modes will result in a slower text entry rate unless the mobile phone user is skilled in using the predictive method. Instead, providing an automatic switch between multi-tap and the predictive software would enable the users to text at a faster rate.

Another notable flaw of the predictive software is the inability to auto-save words that are created by compounding existing words. For instance, the word 'roomie' is not readily available in the T9® dictionary, but the word 'room' is. A user would then usually type the word 'room' first, and then compound an 'ie', thus creating 'roomie'. However, if the user wishes to type 'roomie' again later, he/she would have to retype 'room' followed by compounding 'ie', as T9® does not auto-save compounded words [9].

SMS LIMITATIONS

Apart from the improper keypad layouts for text entry, another of the major limitations of SMS is the overall length of messages allowed, which is fewer than 160 characters per message. An addition of a single character would result in a message length of 161 characters, and hence the message will be split into two. This means that the sender will have to pay for two messages instead of one. When the number of characters exceeds 160, users generally will trace back their message and rephrase the message, deleting some words and abbreviating others so as to reduce the message length. This can be quite cumbersome as the users will have to go through their message again and make the necessary modifications. Furthermore, this process results in a higher message composition time. Due to this, the youth created an ingenious way to message by heavily abbreviating the common words. For example, 'thank you' is typed as 'tq', 'later' as 'l8r', 'already' as 'dy' or 'd', and 'please' as 'pls'. As there is no de-facto standard for the abbreviations, users who are not familiar with such jargons would shy away from using SMS. This is especially true among the older users [10].

Therefore, the aim of this study is twofold: firstly, to enhance the current implementation of the predictive software to enable users to enter messages at a faster rate and hence improve their messaging satisfaction. Secondly, the aim is to provide a facility to abbreviate the messages, both in English and Bahasa Malaysia, the local and national language of Malaysia. When message length is reduced, it is possible to send a lengthier message at the cost of a single message. Moreover, users who are not very familiar in using abbreviations would benefit from this added feature as well.

RELATED WORK

Text Entry

The multi-tap remains to be the prevailing text entry method used on mobile phones, despite being slower than the predictive method [8]. This has led many researchers to explore other techniques to increase text entry speed. For example, many attempted to optimise text entry performance by creating keypad designs that reduce the number of keystrokes needed to enter a word. The one-row keyboard prototype was Nokia's attempt to make the T9® system faster by using more fingers than the normal one or two in typing. The keypad consisted of ten keys, all in one row. The alphabets were similarly distributed among the keys as in T9®. However, tests indicated that the system in fact made the typing of words slower than the T9® system [11].

Sörensen and Springael [12] proposed a new keypad layout to improve text entry when using the predictive method. Called Keymap, it was designed by assigning as many letters as possible on the same key, and then the letters that could cause collisions were distributed on different keys. This keypad works based on Iterated Local Search, an algorithm used to find the best placement of each letter over the keypad keys.

LetterWise [13] is another proposed system to improve text entry using prefix-based disambiguation. It works with a stored database of probabilities of prefixes (letters preceding the current keystroke). For example, if key-3 is pressed with prefix 'th', the most likely next letter is 'e' because 'the' in English is far more probable than either 'thd' or 'thf'.

Some efforts have also been made to move away from text entry optimisation, exploring gestural interactions such as the use of joysticks [14], tilting [15], motion [16] and also the possibility of using speech to text [17].

MessageAbbreviation

It is important to note that message abbreviation differs from message compression. A compression method generally reduces the message length by encoding the message in fuzzy characters, and hence it requires the receiver to decode the characters to its original form. Although this technique successfully reduces the message length, both communicating parties need to have the compressor/decompressor at their ends. Currently, there are very few implementations of SMS compression, one of which worth mentioning is KirimSMS [18]. This program uses the traditional fixed Huffman coding that reduces the code length of frequently used characters while increasing the code length of those that are less frequently used. Messages that are generally split into two are sent as one message based on the compression software. However, as mentioned before, the sender and receiver must have the same software installed on their mobile phones; otherwise the recipient could only view unintelligible characters [18].

Instead of compressing a message, abbreviation of the common words to reduce the overall length of the message is used in this study. The technique used is similar to the abbreviation technique employed in the computer-based MobiSMS for Outlook which was developed by MobiMarketing [19]. This application abbreviates a message based on a dictionary located in the computer hard drive. An unabbreviated word is paired with its abbreviated partner and kept in this dictionary, which the user can easily modify to add more abbreviations, words, or to edit existing ones. For example, the sentence 'Please phone me if you can meet for lunch at one today' will be abbreviated to 'Pls phone me if u can meet 4 lunch @ 1 2day.' However, this program is only available for Outlook, and thus can only be used on a computer [19].

The main aim of this study is to design an application that abbreviates common words in a text message to make the whole message significantly shorter yet understandable, without requiring the users to have the exact application in their mobile phones. Moreover, a lossless abbreviation technique is implemented whereby punctuations, spaces, etc. remain intact even after the abbreviation takes place.

METHOD

For the enhancement of the predictive text system, some of the existing ideas in T9® were retained. These include the dynamic search for words from the dictionary on-the-go as the user types, ability to learn new words (though this is not done automatically), the use of the asterisk key to scroll through the list of possible words, and the use of word frequency to determine the most probable word that the user intends to type. The current study aims to enhance T9® by including the ability to automatically (i) save new words into the dictionary, (ii) learn words by compounding, (iii) insert a question mark when words such as 'how, where and when,' are encountered, and (iv) enter numerals. In short, the system works on a one-mode-for-all concept, whereby users get to perform all the functions above without having to switch to multi-tap, or from multi-tap to predictive mode.

For the abbreviation mechanism, an algorithm that is very close to MobiSMS's was used, whereby unabbreviated words were matched with their abbreviated partners in the word list or stored dictionary. If a particular word has no abbreviated partner in the dictionary or if the entry is already an abbreviated word, then the original word will be displayed as it is.

The general algorithm used in the current system can be presented in four main steps, beginning with the enhancement of the predictive text entry. Figure 4 shows the steps for both the enhancement and the abbreviation. Note that the abbreviator can be used regardless of the text entry mechanism employed.

The four main steps as indicated in Figure 4 can be elaborated as follows:

Step 1: Get the code sequence, e.g. '5299', entered by the user, breaking only if the 'right', 'send', 'delete', 'hash' or 'asterisk' key is detected.

Step 2: Constantly update the list of possible words with words corresponding with the code sequence by checking with the dictionary. Sort the list according to the frequency of each word, with the highest frequency word sorted to the top. For example, '5299' would produce the list of words as depicted in Table 1. To cycle through the list of possible words, the user would have to press the asterisk key. If the intended word did not exist in the used word list or the dictionary, an automatic switch to multitap mode would occur, and the list as well as the word would be cleared to allow a multitap entry of the word (first time entry).

Step 3: Once a right directional key is detected, the word that was previously entered would be saved to the dictionary. If the word already existed in the dictionary, then its frequency (prediction rate) would be increased.

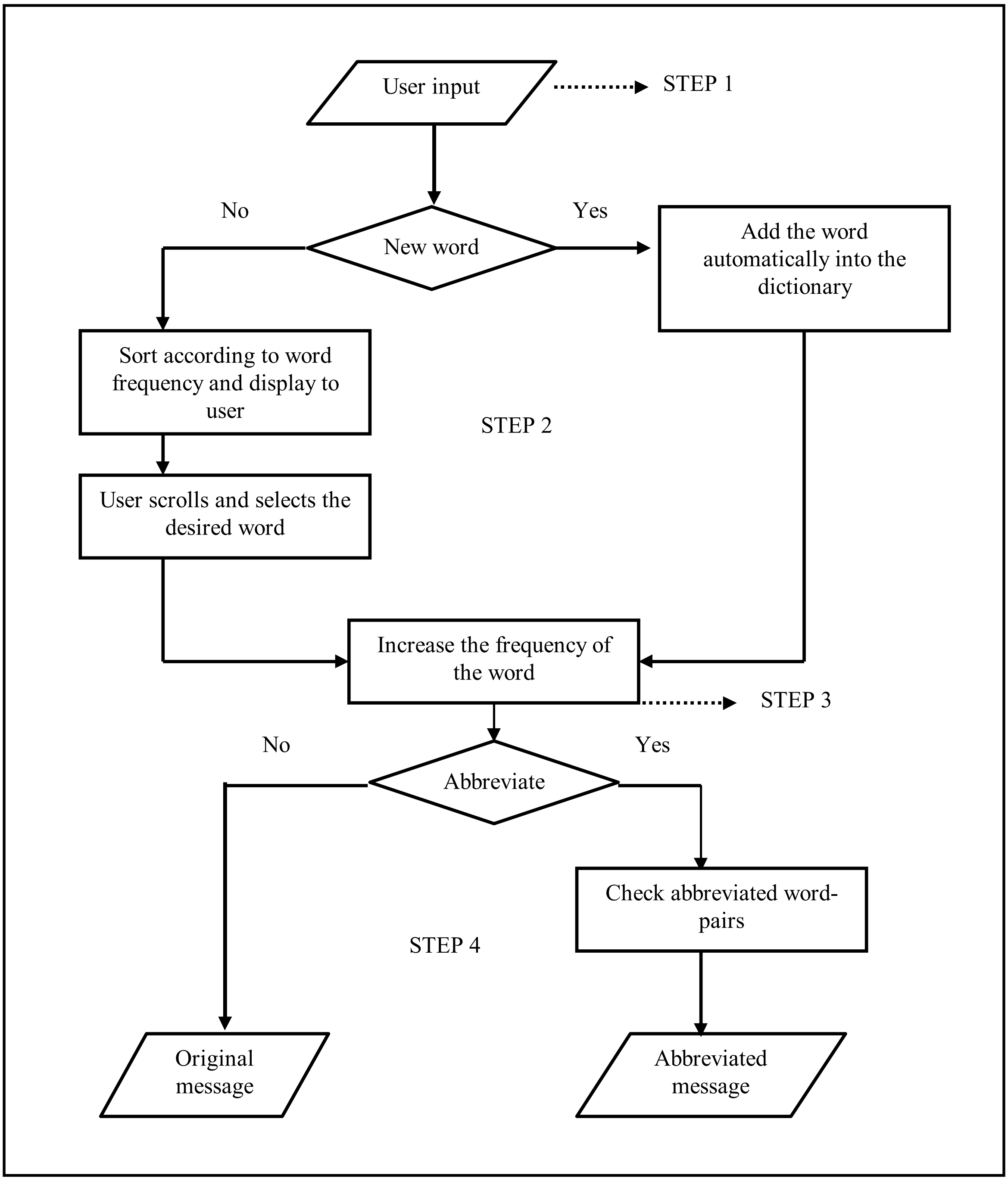


Figure 4. General algorithm for the predictive system and abbreviator

Step 4: Once the user has finished typing the desired message, he/she would be prompted to choose to abbreviate the message. If selected, the abbreviator would iterate through all the words in the original message, find their abbreviations if any, and copy the abbreviation, or if no abbreviation is found, the original word is retained.

Maejo Int. J. Sci. Technol. **2011**, *5*(02), 266-278

Table 1. The list of possible words generated and the code sequence

Code sequence	5	52	529	5299
List of possible words	K, L, J	La, Lb, Ka, Ja, Kb	Jay, Law, Lay, Jaw, Lax, Kay, Laz, Jaz, Kaz	Jazz, Lazy, Lawy

In step 4 above, the abbreviation takes place only when the message is completed and not after the completion of each word in the message. Having a process where the words are abbreviated automatically as they are entered may interrupt the flow of messaging or perhaps even confuse the user.

The system was developed using Java and was simulated on a mobile phone environment using the SERIES 40 SDK Version 0.9 in conjunction with JRE 6.0.

RESULTS AND DISCUSSION

Enhancement of the Predictive Software

Figure 5 shows a simple example involving a question. The number '29/1' at the top right-hand corner means there are 29 characters in the message and the total message is one. The underlined word refers to the current word being entered, which is 'fine'. In this example, the system automatically inserts a question mark when a sentence involving 'how, where and where, etc.' is detected rather than the period character '.'. The question mark will be automatically inserted when the user presses the hash key for punctuation. The current predictive software does not have this ability and users are required to enter the question mark manually.

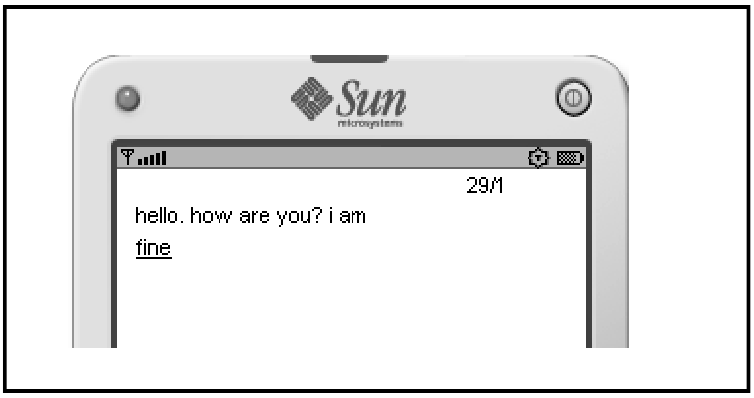
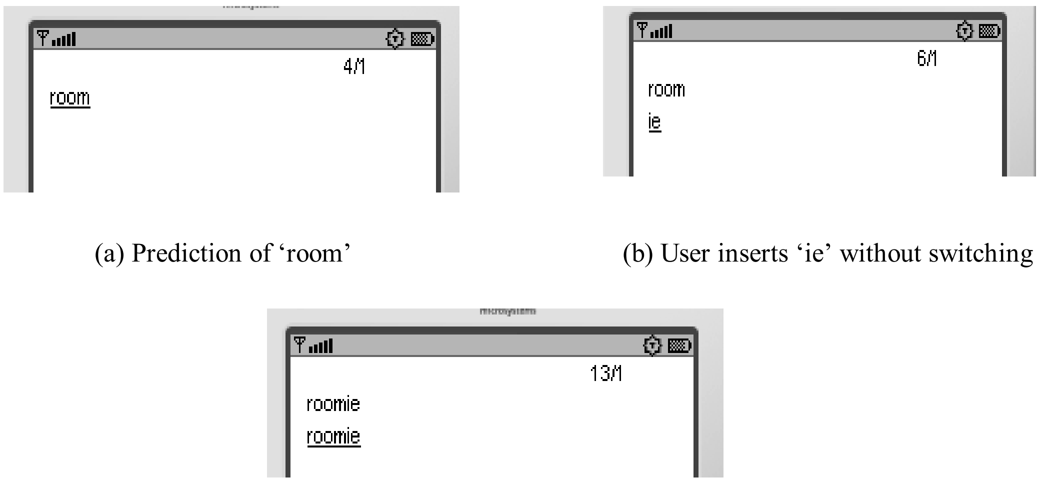


Figure 5. Automatic insertion of the question mark

Maejo Int. J. Sci. Technol. **2011**, *5*(02), 266-278

The system is also capable of automatically adding new words into the dictionary, including compounded words. Users need not switch to the multi-tap mode manually. In other words, the users can enter any new words and they will be saved automatically without the user's intervention. Figure 6 shows an entry example for a new compounded word, 'roomie.' The system will predict 'room' (Figure 6a) and then the user can automatically insert 'ie' (Figure 6b) without any manual mode switching. The word 'roomie' is automatically added to the dictionary, and hence it will be predicted when another attempt to enter 'roomie' is encountered (Figure 6c).



(a) Prediction of 'room'

(b) User inserts 'ie' without switching

(c) Prediction of 'roomie'

Figure 6. Automatic learning of a new compounded word

Similarly, the system also allows the user to easily enter numerals without the need to switch to the number mode, hence making messaging simpler and faster (Figure 7).

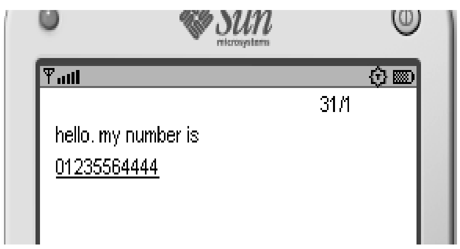


Figure 7. Automatic numeral entry

Maejo Int. J. Sci. Technol. **2011**, *5*(02), 266-278

Abbreviator

This section shows the displays for the abbreviator which can be used regardless of the text entry methods. Figure 8(a) shows the original message as entered by the user. This particular message consists of 178 characters, and hence would be split into two messages, as indicated on the screen. Figure 8(b) shows the message after the abbreviation. Words that are already abbreviated and those that do not have an abbreviation partner are left in their original forms. Note that the message length now is reduced to 157 characters and the number of message is one instead of two. Apart from saving cost, this feature would be beneficial to users who are not so adept in using abbreviations.

(a) Original message

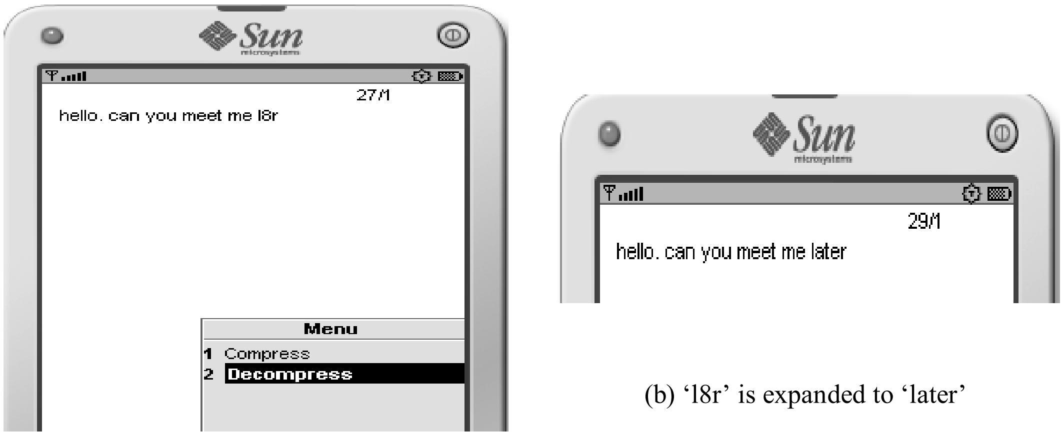
(b) Message after abbreviation

Figure 8. Example of abbreviations

Another advantage of this system is that the recipient can choose to expand the abbreviated word/message to its original form by choosing decompression from the menu (Figure 9a). Figure 9(b) shows how the originally abbreviated word ‘l8r’ is expanded to ‘later’. This feature would be very useful especially if users are unsure about the meaning of an abbreviated word.

Finally, a dictionary to support messages in the national language, i.e. Bahasa Malaysia, was also added into the system. The original message in Figure 10(a) can be translated (non-verbatim) as ‘I will not be able to report to work today, I would like to apply for leave for today and tomorrow.’ Figure 10(b) depicts the abbreviated message consisting of only 91 characters, shorter than the original message with 109 characters.

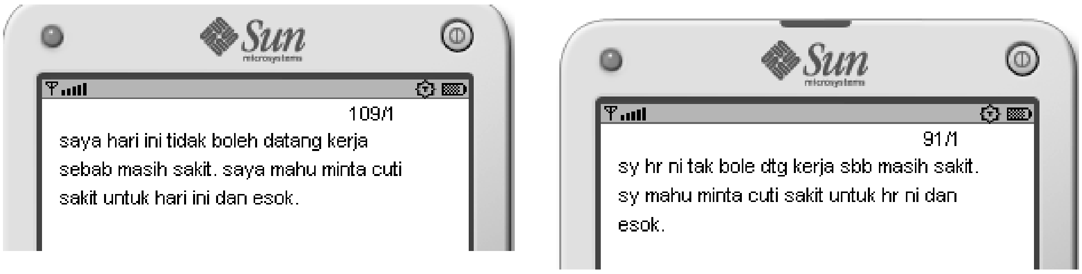
Maejo Int. J. Sci. Technol. **2011**, *5*(02), 266-278



(a) Received message

(b) ‘l8r’ is expanded to ‘later’

Figure 9. Example of expanding abbreviated words to their original forms



(a) Original message – 109 characters

(b) Abbreviated message – 91 characters

Figure 10. Example of abbreviations in Bahasa Malaysia

Limitations and Future Work

The implemented system is not without its limitations. First of all, the system does not support insertion. For example, if the user wanted to enter ‘March’ but accidentally keyed in ‘Marh’, he/she needs to delete the letter ‘h’ to enter ‘ch’. Instead, it would be simpler if the user could just move to the intended location and insert the missing character(s). Currently this can only be done on mobile phones with QWERTY keyboard.

As for the abbreviator, new word-pairs currently cannot be added into the dictionary, meaning users only depend on the existing pairs to abbreviate their messages. Future work would provide the flexibility for users to add their own word-pairs. Another important feature that will be included is the support for emoticons (for mobile phones that support these icons), since users, especially the youngsters, have a high tendency to use emoticons in their messages.

Maejo Int. J. Sci. Technol. **2011**, *5*(02), 266-278

It should be noted that the system was tested in a mobile simulated environment. Therefore, though it is inferred that the system should work successfully when implemented in the actual mobile phones, further study is required to conclusively confirm the efficacy of the system. The implementation will be ported to real mobile phones for further testing in future work.

Finally, the future work will also include a usability study. This is necessary to assess mobile phone users' messaging satisfaction when using the implemented system. One of the features that will be tested is the option given to the users for abbreviation. Currently, it is assumed that automatic abbreviations (as user types the message) would confuse the users; therefore, an option to abbreviate or not is provided. User testing with (and without) the option needs to be conducted to address this assumption in the future.

CONCLUSIONS

In this study, a one-mode-for-all application for the predictive text entry software on mobile phones was developed. The existing predictor software was enhanced by enabling automatic word saving (both new and compounded words), insertion of question marks and automatic entry of numerals. All of these features can be done without having to switch to multi-tap, thus saving message composition time. In addition, an abbreviator that abbreviates and expands messages both in English and Bahasa Malaysia was also implemented. This would be useful as users who are not adept in using abbreviations can now do so. Moreover, message lengths will be reduced and hence users can compose and send longer messages at the cost of a single message (as long as the total characters are less than 160).

REFERENCES

1. J. Sciallo, "Short message service (SMS) support in Sun Java System Messaging Server 6.3", http://www.sun.com/bigadmin/features/hub_articles/sms_support_msg_server.jsp, **2007** (accessed: November 30 2008).
2. S. N. Audrey and L. Srivastava, "Mobile phones and youth: A look at the U.S. student market", ITU/MIC Workshop on Shaping the Future Mobile Information Society and Policy Unit, **2004**, Seoul, South Korea.
3. Maxis Co., "Value Plus 50", http://www.maxis.com.my/personal/mobile/plan_pricing/value_plan_50.asp?iStruct=0:2:1, **2009** (accessed: February 11, 2010).
4. T. Igarashi, J. Takai and T. Yoshida, "Gender differences in social network development via mobile phone text messages: A longitudinal study", *J. Social Person. Relations.*, **2005**, *22*, 691–713.
5. L. Leung, "Unwillingness-to-communicate and college students' motives in SMS mobile messaging", *Telemat. Informatics*, **2007**, *24*, 115–129.
6. D. L. Grover, M. T. King and C. A. Kuschler, "Reduced keyboard disambiguating computer", Tegic Communications, Inc., *US5818437* **(1998)**.

Maejo Int. J. Sci. Technol. **2011**, *5*(02), 266-278

7. I. S. MacKenzie and K. Tanaka-Ishii, "Text entry using a small number of buttons", in "Text Entry Systems: Mobility, Accessibility, Universality" (Ed. I. S. Mackenzie and K. Tanaka-Ishii), Morgan Kaufmann, San Francisco, **2007**, pp.105–121.
8. M. Silfverberg, I. S. MacKenzie and P. Korhonen, "Predicting text entry speed on mobile phones", Proceedings of the ACM Conference on Human Factors in Computing Systems, **2000**, New York, USA, pp.9–16.
9. S. Rajeev, S. Ghosh, G. N. Kumar and S. N. Sivanandam, "A simple algorithm for text entry in mobile communication devices", *Acad. Open Internet J.,* **2003,** *10*, http://www.acadjournal.com/2003/v10/part1/p1/, (accessed: December 11, 2008).
10. H. Du and F. Crestani, "Spoken versus written queries for mobile information access", in "Mobile and Ubiquitous Information Access - Mobile HCI 2003 International Workshop", (Ed. F. Crestani, M. Dunlop and S. Mizzaro), Springer-Varlag, Udine, **2004**, pp.67–78.
11. M. Silfverberg, 'The one-row keyboard: A case study in mobile text input', in "Mobile Usability: How Nokia Changed the Face of the Mobile Phone" (Ed. C. Lindholm, T. Keinonen and H. Kiljander), McGraw Hill, New York, **2003**, pp. 157–168.
12. K. Sörensen and J. Springael, "Optimization of SMS keyboards with a dictionary using local search", Workshop on Real Life Applications of Metaheuristics, **2003**, Antwerp, Belgium.
13. I. S. MacKenzie, H. Kober, D. Smith, T. Jones and E. Skepner, "LetterWise: Prefix-based disambiguation for mobile text input", Proceedings of the ACM Symposium on User Interface Software and Technology, **2001**, New York, USA, pp. 111–120.
14. J. O. Wobbrock, B. A. Myers and H. H. Aung, "Writing with a joystick: A comparison of Date Stamp, Selection Keyboard and EdgeWrite", Proceedings of Graphics Interface Conference, **2004**, London, UK, pp.1‑8.
15. D. Wigdor and R. Balakrishnan, "Tilttext: Using tilt for text input to mobile phones", *Comp. Inform. Sci.*, **2003**, *5*, 81‑90.
16. J. Wang, S. Zhai and J. Canny, "Camera phone based motion sensing: Interaction techniques, applications and performance study", Proceedings of the ACM Symposium on User Interface Software and Technology, **2006**, Montreux, Switzerland, pp.101– 110.
17. A. L. Cox, P. A Cairns, A. Walton and S. Lee, "Tlk or txt? Using voice input for SMS composition", *Person.Ubiquit. Comput.*, **2008**, *12*, 567–588.
18. Y. Sugianto, "SMS text compression", http://www.kejut.com/kompresisms, **2007** (accessed: April 10, 2010).
19. T. D. Susanto and R. D. Goodwin, "Content presentation and SMS-based e-government", Proceedings of the International Conference on Electrical Engineering and Informatics, **2007**, Jakarta, Indonesia, pp.1082–1085.