LuitPad: A fully Unicode compatible Assamese writing software

Navanath Saharia[1,3] *Kishori M Konwar*[2,3]
(1) Tezpur University, Tezpur, Assam, India
(2) University of British Columbia, Vancouver, Canada
(3) NavaPrabhat, Tezpur, Assam, India
`navanathsaharia@gmail.com, kishori@engr.uconn.edu`

Abstract

LuitPad is a stand-alone, fully Unicode compliant software designed for rapid typing of Assamese words and characters. There are two main typing options; one which is based on approximate sound of words and the other based on the sound of characters, both of which are efficient and user-friendly, even for a first-time user. In addition, LuitPad comes with an online spell-checker; on "right-clock" over a misspelt word, presents the user with a list of relevant appropriate corrections for replacements. Assamese is an Indic language, spoken throughout North-Eastern parts of India by approximately 30 million people. There is a severe lack of user-friendly software available for typing Assamese text. This is perhaps the underlying reason for the miniscule presence of Assamese based information storage and retrieval systems, both off-line and on-line. With LuitPad, the user can retrieve Assamese characters and words using an English alphabet based keyboard in an effective and intuitive way. LuitPad is compatible with Windows, Mac and Linux with a GUI. The software can store the contents in LuitPad file format (".pad" extension) that can store images and text. In addition, .pad files can be easily exported to pdf and html files.

Keywords: LuitPad, Assamese language, Unicode, text input.

*Corresponding author

1 Introduction

Assamese is one of the commonly spoken languages in the North-Eastern parts of India. Approximately 30 million people speak Assamese, and it is the regional official language in the state of Assam. Although Unicode points have been assigned for the Assamese alphabets Assamese texts on the Internet or in digitized format is almost non-existent. There are several hundred commonly used glyphs related to Assamese characters. At present, there is no text editing software to write Assamese texts in an effective manner. The most popular software, commonly used in desktop publishing shops and newspaper printing houses, is a non-Unicode based software. The software uses non-Unicode proprietary font files and reuses the ASCII code points to render non-English glyphs. In order to circumvent the limitation of just 127 ASCII codes, a set of segmented strokes are represented using ASCII codes in a font file. For example, in an ordinary font file the glyph at the hexadecimal code 0x41 is the shape for the letter “A” but in the custom designed font file it may look like something very different. Most of the characters are rendered by juxtaposing 2-3 of these strokes. In order to accommodate bold, italic and Assamese numerals there are other separate font files. Hence, an ASCII code can correspond to several different glyphs depending on the font file. As a result, typing characters involves pressing complex key combinations. This also confines inputting of Assamese texts on computers only to highly trained professionals. Due to the above mentioned difficulties and lack of effective Assamese text editing software Assamese text on the Internet, or any other digitized form, is very rare. Among the very few websites that use Assamese texts, mostly daily newspapers, the overwhelming majority presents their contents in the form of images of pages. This prohibits text searching and mining in Assamese documents.

LuitPad is developed to address the above difficulties of Assamese text editing on computers. It is designed to be able to edit Assamese text in Unicode by novice users with minimal effort and training. The Unicode complaint nature of LuitPad allows the user to copy Assamese text from the LuitPad editor to any email in the Unicode format, and view it elsewhere, with commonly used web-browsers (e.g. FireFox, Google Chrome, Internet Explorer, Safari, etc). In addition, approximate pronunciation based mode in the software prevents the user from making potential spelling mistakes by presenting a list of possible words. The interactively presented words are retrieved from an internal dictionary, with more than 60,000 Assamese words, which also includes proper nouns. Most Assamese dictionaries list about 30,000-40,000 words. The efficient retrieval of words is done by using elegant rapid searching data-structures and morphological trimming algorithms developed by the authors. In addition, the approximate pronunciation based search algorithms in LuitPad are tailored to commonly used Assamese.

The rest of the paper is organized as follows. In section 2 we describe previous work related to Assamese script and writing tools. Section 3 explains the features of LuitPad software; Section 4 discusses the experimental tests of the software; and finally, Section 5 concludes the paper.

2 Background on Assamese writing and previous work

Very few studies have been reported on the computational linguistic approaches of the Assamese language (Saharia et al., 2010, 2012; Sharma et al., 2002, 2008) and Assamese writing systems (Hinkle et al., 2010).

Brief description of the Assamese script Assamese writing has evolved from the ancient Indian script, *Brahmi*. Brahmi had various classes, and Assamese originated from Gupta Brahmi Script. The history of Assamese script and language can be divided into three main era, ancient, medieval and modern script. During each era, the language went though significant changes in terms of script, language and literature. The oldest Assamese inscriptions were found in the state of Assam are 5th century rock inscriptions at *Umachal* in *Guwahati*[1] and the *Nagajuri Khanikargaon inscription*, in Golaghat district. These two samples of inscription are in Sanskrit (Brahmi script). The *Kanai Barasi Bowa rock inscription* at *Rajaduar* in *north Guwahati*, dating back to 1205-06 AD, marks the first stage of evolution of the Assamese alphabet and deviation from Brahmi script. Assamese alphabets consist of 11 vowels, 41 consonants, 143-174 two-phoneme and 21-27 three-phoneme clusters of conjunct letters. Table 1 tabulates Assamese consonant and vowels and Table 2 shows digits, example of a consonant with example usage of vowel modifiers and a few instances of multiple cluster letters.

Consonants				
ক	খ	গ	ঘ	ঙ
চ	ছ	জ	ঝ	ঞ
ট	ঠ	ড	ঢ	ণ
ত	থ	দ	ধ	ন
প	ফ	ব	ভ	ম
য	ৰ	ল	ৱ	
শ	ষ	স	হ	
ক্ষ	ড়	ঢ়	য়	
ৎ	◌ং	◌ঃ	◌ঁ	

Vowels				
অ	আ	ই	ঈ	
উ	ঊ	ঋ		
এ	ঐ	ও	ঔ	

Table 1: Assamese consonants and vowels

Digits						
০	১	২	৩	৪	৫	৬
৭	৮	৯				

Consonant ক with vowel modifiers						
ক	কা	কি	কী	কু	কূ	কৃ
কে	কৈ	কো	কৌ			

Some two cluster letters						
ক্ক	ক্ল	ক্ত	ক্ম	ক্স	স্ম	প্ত
স্ক	ঙ্ক	স্ত	ঙ্গ	ন্ত	ত্ত	ন্ন
ল্ল	ষ্ণ	শ্ত	হ্ল	স্ন	ষ্ট	প্প

Some three cluster letters					
ত্ত্ব	ন্ধ্র	স্ত্র	জ্জ্ব	ম্ভ্র	ন্ধ্য

Table 2: Digits, consonants with vowel modifiers and, examples of 2 & 3 letter clusters.

Brief description of previously used techniques When we discuss text inputting software, usually the focus is on the soft-keyboards and other methods/techniques such as auto-completion, character/word prediction, handling morphology and word list etc. Due to the large (in the order of several hundreds) set of letters and conjunct letters (*cluster letters*), in comparison to the English alphabets, writing Assamese with an English keyboard is extremely inefficient, The Bureau of Indian Standards adopted ISCII[2] to represent text in Indian languages. After that, Unicode[3] adopted the ISCII with some modifications. Among the Indian languages Hindi, Bengali and Telugu are explored more

[1]The capital of the state of Assam and the gateway of North-Eastern India

[2]Indian Standard Code for Information Interchange (ISCII) is an 8-bit character encoding, for Indian languages originate from Brahmi script. ISCII is adopted as a standard by Bureau of Indian Standards (BIS) in 1991.

[3]An UNIversal enCODEing system, was initiated in January 1991, under the name Unicode, Inc., to promote a common encoding system to process text of all languages. Currently, Unicode is in version 6.1.0. It provides three different encoding formats: UTF-8, UTF-16 and UTF-32.

than other Indian languages. For soft-keyboards, only one Assamese specific work has been reported in (Hinkle et al., 2010). They proposed an efficient and adaptable soft keyboard for Assamese, using simple genetic algorithms. They also reported techniques for creating optimal language independent soft-keyboards. The product is not yet available in the market. Google(*iGoogle*) has a predictive soft-keyboard for Assamese. In this static keyboard, alphabets are arranged in the boundary of the input box. It has lesser number of keys since both the vowels and phonetic variant buttons are missing from the layout.

During our literature survey we found only a few tools available for Assamese text writing. Among them *Ramdhenu*[4], an ASCII based Windows tool is more popular in desktop printing businesses and new-papers. It is a hooking application that interfaces with the active application (for example MS Office, Notepad etc.) and change the font mapping to its own font *Geetanjali*. In the Geetanjali, the ASCII codes are assigned some carefully chosen strokes and segments in a way that the glyph for each character can be created by collating these segments. The main disadvantage of this tool is it is ASCII based, and difficult to input Assamese digits and cluster characters. It requires (*Ctrl* + *Shift*) key press combinations to input numerals. The user has to change font face in order to render italic or bold text. On the other hand, Avro (OmicronLab, 2012), Baraha (Baraha, 2012), Lipikaar (Lipikaar, 2012) and Sabdalipii (Sabdalipi, 2012) are Unicode based writing tools, the first three are for all Indian languages and final one is specifically for Assamese. Avro and Baraha supports phonetics based writing but requires quite a large number of key strokes. Baraha, Sabdalipi and Lipikaar have integrated text editors. Google and Wikipedia have developed soft keyboards for Brahmic scripts, but these are not optimized(Hinkle et al., 2010), and therefore, inefficient to use.

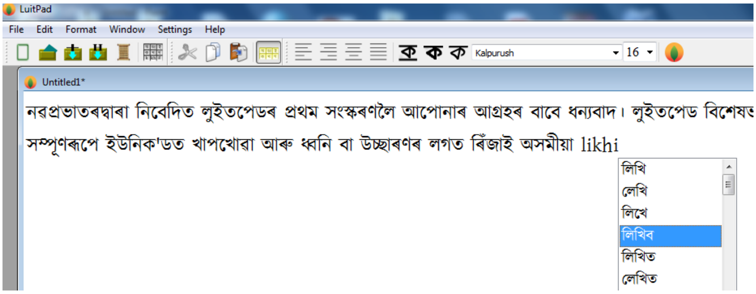
3 Description of LuitPad

LuitPad is designed to make inputting of Assamese text efficient in the Unicode format, by lay users and professional typists. LuitPad achieves this by allowing the user to retrieve Assamese characters and words, with English alphabet based keyboards, in an effective and intuitive way. This helps the user to use LuitPad effectively, with less than 20 minutes of training. In addition, it prevents the user from making potential spelling mistakes by utilizing an internal dictionary of more than 60,000 Assamese words including proper nouns. In the remainder of the article, by "character" we mean Assamese characters and when we refer to English or Roman alphabets we do so explicitly. All the keys and keyboards we refer to are based on the Roman or English alphabet. Following are the fundamental features of LuitPad that are pertinent to text inputting.

1. LuitPad has two primary modes of input: *character* mode, where characters are entered one at a time; and the *word* mode, where user can input Assamese words by typing an "approximate" phoneme of the Assamese word, in Roman alphabets.

2. In the character mode, each Assamese alphabet is mapped from a prefix string of Roman alphabets (usually, 1-3 characters long). The mapping is customizable by the user to suit her preferred way of pronouncing Assamese characters, using Roman alphabets. Multiple users can store, retrieve and change their mapping in their personalized profiles, in the same installation of LuitPad.

[4]Ramdhenu Inc.



নৱপ্ৰভাতৰদ্বাৰা নিবেদিত লুইতপেডৰ প্ৰথম সংস্কৰণলৈ আপোনাৰ আগ্ৰহৰ বাবে ধন্যবাদ। লুইতপেড বিশেষত
সম্পূৰ্ণৰূপে ইউনিক'ডত খাপখোৱা আৰু ধ্বনি বা উচ্চাৰণৰ লগত ৰিঁজাই অসমীয়া likhi

Figure 1: The front end GUI of LuitPad, in the Windows version, with a editor window showing a few lines of Assamese text and an image.

3. In the character mode, most of the Assamese words can be retrieved on the text editor by just typing the first few characters and pressing the *Ctrl* key. This brings up a list of words from an internal built-in Assamese dictionary. Apart from not having to type the entire word, this also reduces the possibility of making spelling mistakes.

4. LuitPad is also equipped with a spell checker for Assamese words and prompts a list of relevant corrections for spelling mistakes.

5. LuitPad text editor allows inserting, positioning and resizing of images.

6. Documents edited using LuitPad can be saved as a .pad file; and also can be edited later using LuitPad. The .pad file format is designed specifically for LuitPad.

7. Files created with the LuitPad text editor can be saved in PDF and HTML format for printing or other official works.

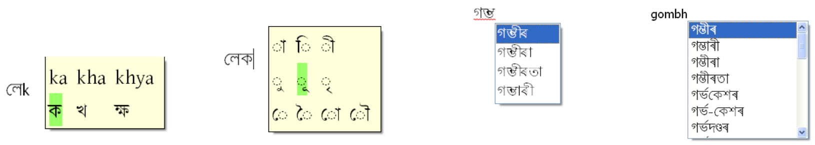
8. New words can be added to the built-in dictionary, while writing, simply by right clicking the mouse and choosing the “Add to dictionary” option; and similarly, already added words can be deleted by using the “Delete from dictionary”. The added words can also be recalled using the *Ctrl* key as described above.

9. Any document written in the Geetanjali (a popular non-Unicode format for Assamese) font can be seamlessly converted to and fro to Unicode format by copy/paste.

10. LuitPad is a stand software, 100% Unicode compliant, and runs on Windows, Linux and Mac computers.

3.1 Input modes in more detail

LuitPad has the text input modes: *character* and *word* modes. The selected texts are always entered at the cursor. Below we describe these two modes in more detail.



(a) (b) (c) (d)

Figure 2: (a) In character mode a pop-up presents the set of possible characters to choose from. (b) A vowel modifier list pop-ups, if the "Automatic Matra" option is turned on, in the character mode after non-vowel character. (c) Prefix based recalling of dictionary words in character mode. (d) In word mode, recalling dictionary words with approximate pronunciation in Roman alphabets.

Character mode: In this mode, each character in the Assamese alphabet, just consonants, vowels and vowel modifiers but not conjuncts, is mapped to a short string of Roman alphabets (usually, 1-4 characters but there is no length limit and they are customizable by the user). For example, the characters ক, খ and ক্ষ are mapped by *ka*, *kha* and *khya*, respectively. The digits 0-9 on the keyboard are mapped to the corresponding Assamese digits. When a user types any prefix of the mapped strings, in Roman alphabets, all the corresponding Assamese characters are presented on a tool-tip pop-up window, for the user to choose from. The *left* (←) and *right* (→) keys can be used, followed by <Enter>, to choose the desired character. The chosen character is inserted into the text editor. For example, consider the mapping, mentioned above, for the characters ক, খ and ক্ষ and suppose these are the only characters that have mapped strings beginning with the Roman alphabet "k". Upon typing "k" the the tool-tip window shows ক, খ and ক্ষ If "h" is pressed next, the characters খ and ক্ষ corresponding to the prefixes "kha" and "khya" will appear in the tool-tip window. Therefore, as the user progressively types out a possible mapped prefix the choices get narrower. If the user types out a possible mapping string which is not mapped to any character then the tool-tip window stops appearing. If "Automatic Character Fill" option, under the "Settings" menu is on and the prefix corresponds to only one possible character, then the character is automatically inserted, without requiring the <Enter> key-press. In addition, if the "Automatic Matra" option, in the "Settings" menu, is on and if the entered character is a consonant, a tool-tip window with the possible vowel modifiers is shown. If the user does not want to select a vowel modifier, she can continue entering the next character. In order to reduce typing, an entire word can be retrieved by typing the first few characters followed by a *Ctrl* key press. This will bring up a drop down list of words containing the prefix to choose from. Conjuncts can be created by pressing <Escape>, with the cursor at the end of a juxtaposition of the consonants.

Word mode: In the *word mode* entire Assamese words can be inserted just by typing an approximate phoneme for the Assamese word. The "approximate" phoneme helps accommodate different phonetic sounds for the same word, by different users. For example, the phonemes "karma", "korma", "kormo" and "karmo" might be used by different users to retrieve same word কৰ্ম. A dynamically changing drop down list helps the user to navigate to

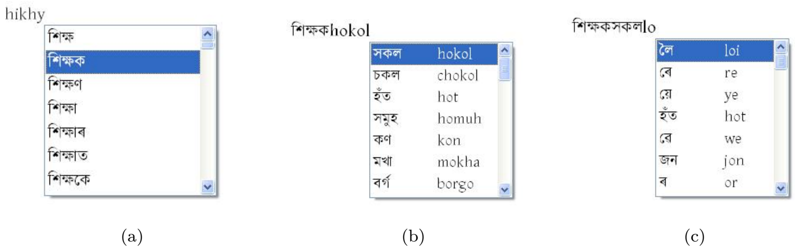


Figure 3: The stages of typing a word with morphological inflections. (a) Root word, (b) first inflection and (c) second inflection

the correct word, to deal with possible ambiguity. The list is updated with the completion of each key stroke. Once the user selects a word from the list, using the up(↑)/down(↓) keys, the selected word is inserted at the cursor. After the user enters a word then if the user presses <Space> bar then the process of inserting a separate word starts. However, if the user continues typing, without pressing <Space> bar, then the key strokes are interpreted to a get list of closest matching morphological inflections. It is worth mentioning here that Assamese is a highly inflectional language, and a root word may have up to 1500 inflectional forms. Consider the word শিক্ষকসকললৈ which can be broken into the root word, followed by morphological inflections as, শিক্ষকসকললৈ $\rightarrow$ শিক্ষক (root word) + সকল (inflection) + লৈ (inflection). The sequence of selections leading to the entering of the entire word শিক্ষকসকললৈ is shown in Figure 3. In practice, we observed a substantial improvement in the efficiency of Assamese text typing by common users. The users could type up to 15-20 words per minute in LuitPad as opposed to 5-8 words per minute using other available software.

Spell checker LuitPad has a spell checker for Assamese words. Based on its internal dictionary the words with incorrect spellings are flagged with a red under line. The user can choose to ignore any spelling error marks by clicking the ignore tab. The list of words, deemed spelling errors, are ignored in the document and does not affect other documents. This ignore list is stored as a part of the data-structure of the ".pad" file. However, in order to add a word to the ignore list to all documents the the unrecognized word should be added to the dictionary. The basic mechanism of the spell checking feature is done by computing the Levenshtein distance between a word and the words in the dictionary. Note this pairwise distance computation can introduce a perceptible delay in the distance computation. Therefore, we have created an algorithm and a data-structure that avoids such pair wise but rather does a directed search in the dictionary. Although, a morphologically inflected form of a word may not be in the dictionary we use a stemming algorithm to compute the list of root words, at most 4-5, to do the search.

3.2 Algorithms and data curation

The most commonly used algorithms, in the software, are various forms of basic tree based algorithms commonly presented in any data-structure text book. The Levenshtein distance,

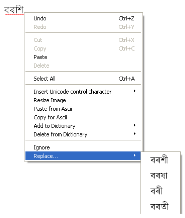


Figure 4: The LuitPad spell checker, with its presented correct choices, for a misspelt word. In the above case the top choice is the correct choice. The figure shows only the top five choices.

between two words, calculations are done using dynamic programming algorithms, implementation by the authors. The penalty matrix for the distance computation algorithms are based on linguistic proximity of the characters. For example, two similar sounding vowels are assigned a lower penalty value. The internal dictionary used, in its current form is only a list of Assamese words. Some of the words are with morphological inflections. The list of words are collected for a variety of sources, such as online dictionaries, bloggins sites. The gathered words are first programatically filtered for basic inconsistencies and later curated one by one. At its current state there are slightly more than 75,000 entries in the dictionary.

4 Experiments with users

Since its inception, LuitPad has been designed, and often redesigned, based on the feedback from various users. Some features were added based on the average users' preferences and others were deleted. We also conducted a quantitative test on the software. The authors are not aware of any benchmarking text dataset, designed for evaluation of typing speed, for the Assamese language. We evaluated the words per minute typing speed, for a group of 14 users by letting them type from commonly printed text. Half of the users do not write Assamese text language on a regular basis, but they can read very well The mean reciprocal rank(MRR), for the retrieval of list of words in the phonetic mode, is computed as the average of the reciprocals of 100 word searches from a popular magazine. Words that did not show up in among the, mostly proper nouns, were ranked infinity, the reciprocal of which is taken as zero. The accuracy of the spell checker is computed based on the percentage of appearance of the correctly spelt word, among the list of suggestions (a maximum of 10 suggestions are presented by the software).

#Users	Words per min	Spell checker accuracy	MRR
14	8-22	93%	0.76

Table 3: Results of the experiments on accuracy and speed of LuitPad.

5 Conclusion

In this paper we described a software, LuitPad, developed for Assamese text writing, in an efficient way. Future work will involve including it to other Indic languages, such as, Oriya, Bengali, Manipuri, Hindi, Marathi, etc. This because these languages have a similar origin and overlapping features in terms of character, conjuncts and vowel modifiers. In most of the cases in order to support one of the above Indian languages, the major changes would be to update the list of Unicodes for their glyphs, phonetic sound of alphabets and a reasonable dictionary. In addition, the internal parameters of the search algorithms used will be calibrated automatically to the user.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable feedback and suggestions for improvements. The authors would like thank Maura Hickey for her critical suggestions on the overall content of the paper. Finally, they would like to thank Lovleena Konwar, Sudipto Konwar, Jahnobi Konwar, Dharitri Borgohain and Jyoti Konwar for curating the internal dictionary and testing the software; and Thapelo Vundhla for perusing the final version of the draft and making valuable suggestions.

References

Baraha (2012). Accessed: 2/10/2012.
URL `http://www.baraha.com`

Hinkle, L., Lezcano, M., & Kalita, J. (2010). Designing soft keyboards for Brahmic scripts. In *Proceedings of 8th International Conference on Natural Language Processing*. India: Macmillan Publishers.

Lipikaar (2012). Accessed: 2/10/2012.
URL `http://www.lipikaar.com`

OmicronLab (2012). Accessed: 2/10/2012.
URL `http://www.omicronlab.com`

Sabdalipi (2012). Accessed: 2/10/2012.
URL `http://www.tiassam.com/sabdalipi`

Saharia, N., Sharma, U., & Kalita, J. (2010). A suffix-based noun and verb classifier for an inflectional language. In *Proceedings of the 2010 International Conference on Asian Language Processing*, IALP '10, (pp. 19–22). Washington, DC, USA: IEEE Computer Society.
URL `http://dx.doi.org/10.1109/IALP.2010.64`

Saharia, N., Sharma, U., & Kalita, J. (2012). Analysis and evaluation of stemming algorithms: a case study with Assamese. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, ICACCI '12, (pp. 842–846). New York, NY, USA: ACM.
URL `http://doi.acm.org/10.1145/2345396.2345533`

Sharma, U., Kalita, J., & Das, R. (2002). Unsupervised learning of morphology for building lexicon for a highly inflectional language. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning*.

Sharma, U., Kalita, J. K., & Das, R. K. (2008). Acquisition of morphology of an indic language from text corpus. *ACM Transactions of Asian Language Information Processing (TALIP)*, *7*(3), 9:1–9:33.
URL `http://doi.acm.org/10.1145/1386869.1386871`