Shahjalal University of Science & Technology

Department of Computer Science & Engineering

CSE 452

Ekushe Keyboard: An AI based Bangla Keyboard for Desktop Environment

Authors :

Asif Mahmud
Reg No.:2014331043
4^{th} year, 2^{nd}

Neamul Ahasan Shovon
Reg No.:2014331020
4^{th} year, 2^{nd}

Supervisor:
Biswapriyo Chakrabarty
Assistant Professor, Dept of CSE

Ekushe Keyboard: An AI based Bangla Keyboard for Desktop Environmente

A Project submitted to the Department of Computer Science and Engineering, Shahjalal University of Science and Technology, in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering.

By

Asif Mahmud
Reg. No.: 2014331043
4^{th} year, 2^{nd} Semester

Department of Computer Science and Engineering

Neamul Ahasan Shovon
Reg. No.: 2014331020
4^{th} year, 2^{nd} Semester

Department of Computer Science and Engineering

Supervisor
Biswapriyo Chakrabarty
Assistant Professor
Department of Computer Science and Engineering

February 13, 2019

Recommendation Letter from Supervisor

The project entitled *Ekushe Keyboard: An AI based Bangla Keyboard for Desktop Environment*
submitted by the students

1. Asif Mahmud
2. Neamul Ahasan Shovon

is under my supervision. I, hereby, agree that the project can be submitted for examination.

Signature of the Supervisor:
Name of the Supervisor.: Biswapriyo Chakrabarty
Date: February 13, 2019

Certificate of Acceptance of the Project

The project entitled *Ekushe Keyboard: An AI based Bangla Keyboard for Desktop Environmente*
submitted by the students

1. Asif Mahmud
2. Neamul Ahasan Shovon

on February 13, 2019
is, hereby, accepted as the partial fulfillment of the requirements for the award of their Bachelor Degrees.

Head of the Dept. M. Jahirul Islam PhD., PEng. Professor & Head Department of CSE	Chairman, Exam. Committee M. Jahirul Islam PhD., PEng. Professor & Head Department of CSE	Supervisor Biswapriyo Chakrabarty Assistant Professor Department of CSE

Contents

Abstract i
Acknowledgement i
Table of Contents iv

1 Introduction **1**
1.1 Computer Keyboards 1
1.2 Use of Computer Keyboards 1
1.3 Virtual Keyboards 2
1.4 Phonetic Keyboards 4
1.5 The Idea of Transliteration Keyboard 5
1.6 Why Transliterated Keyboard 6

2 Previous Works **8**

3 Methodology **10**
3.1 Input 10
3.2 System Library 11
3.3 System Library 12
3.4 Bengali Parser 12
3.5 output 15
3.6 Bilingual Switching 15
3.7 Word Suggestion 15
3.8 Word Prediction 18
3.8.1 Data 18
3.8.2 Data preprocessing 18
3.8.3 Approaches 19
3.8.4 N-grams 20
3.9 System Diagram 21

4 System design **22**

5 Data Collections **26**

6 Problems **27**

7 Conclusion **28**

List of Figures

1.1 Keyboard Structure 2
1.2 Virtual Keyboard Structure 3
1.3 Layout 1 7
1.4 Layout 2 7
1.5 Layout 3 7
1.6 Layout 4 7

3.1 Key-mapping Example 1 13
3.2 Key-mapping Example 2 14
3.3 Prefix tree 17
3.4 Prediction comparison LSTM 19
3.5 Prediction comparison N-grams 20
3.6 System diagram 21

4.1 GUI sample 1 23
4.2 GUI sample 2 23
4.3 GUI sample 3 24
4.4 GUI sample 4 25
4.5 GUI sample 5 25

List of Tables

3.1 Notable methods 17
3.2 Data information 18

Abstract

There more than 250 million Bangla native speakers in the world. In this modern age of technology, everyone almost everyone has knowledge of basic computing. So, it became a necessity to provide a system for writing in Bangla in computers. We have developed a project for CSE-452 named EKUSHE KEYBOARD (Desktop). Our development started about ten months ago. In this report we present a detailed description of our development till now of our project, EKUSHE KEYBOARD for Desktop. A detailed methodology, necessity of this project, environments, problems, found solutions, and other possibilities are described in this report.

Keywords: Ekushe, Desktop, Parser, Trie, Transliterated, Keyboard, Suggestion, Prediction, ngram, LSTM.

Acknowledgements

Firstly, we would like to express our sincere gratitude to our supervisor Biswapriyo Chakrabarty, Assistant Professor, Dept. of CSE, SUST, for his continuous guidance and advices. His advices were very helpful and effective. Also, whenever we faced any problems or any difficulties he was always there to help and inspire us.

We are also in debt of our seniors of 2013 batch Gautam Chowdhury and Buddha Chandra Banik. Without their kind help and advice, it was not possible for us to do what we have done till now. They have helped us in every possible way.

Chapter 1

Introduction

1.1 Computer Keyboards

In case of computer interactions with user, keyboard is a typewriter-like device which consist of a number of sequenced keys or buttons. These keys or buttons acts like electronic switches or mechanical levers. After the invention of tele-printer like keyboard it became the main input device for computers leaving behind the punch cards and paper tape.

1.2 Use of Computer Keyboards

In case of general use keyboard is used as a text entry device for word processors, inputting text and numbers, or interacting with any other software in an operating system. Modern computers that are available on the market now-a-days, the meaning and working method of keypresses are independent. A computer keyboard differentiates each discrete key from every other key and represents all key presses to the processing software. Keyboards are also

used for PC gaming either normal keyboards or keyboards with extra special gaming features, which can accomplish most used keystroke combinations. A keyboard is also used to write commands for the operating system of a computer, like as Windows Ctrl-Alt-Delete combination, which provides the operating system general tweaks options screen. A command-line user interface (like Windows CMD) is a type of user interface controlled entirely using a keyboard, or some other input device that does the job of keyboard.

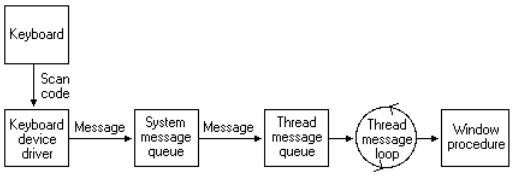

Figure 1.1: Keyboard Structure

1.3 Virtual Keyboards

On the other hand, a virtual keyboard is a piece of abstract software that allows the input of characters without the demand for physical keys. The interaction with the virtual keyboard occurs frequently via a touchscreen device but can also be found in a different form in virtual or augmented reality type devices.

On a desktop computer, a virtual keyboard is used as a way of alternative input system for users with problems and disabilities who cannot use a regular mechanical keyboard, users with the knowledge of multiple language and need to switch quickly and frequently between different language modes, sets or alphabets, which might seem confusing over time, but it's a necessity for them. Although most of the physical keyboards are now-a-days available with dual or multiple keyboard layouts the on-screen keyboard provides a

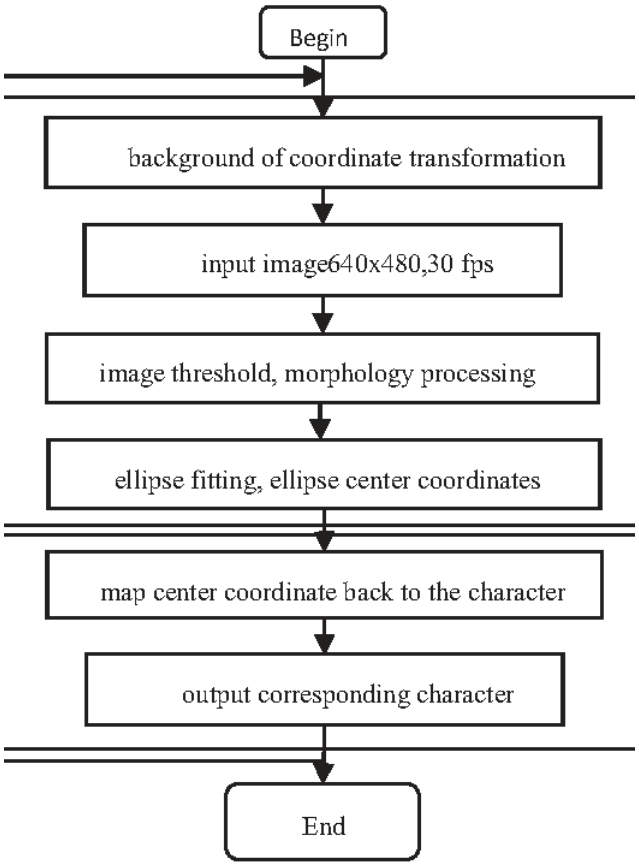

Figure 1.2: Virtual Keyboard Structure

handy alternative while working at different device, mobiles or on laptops, which hardly come with multiple layouts.

1.4 Phonetic Keyboards

A phonetic keyboard layout is a virtual configuration of regular keyboard in which the letters of a language is mapped to the keys in the keyboard layout for different language and assumes a one-to-one correspondence between letters in the languages that is based on their pronunciation.

A transliterated keyboard, on the other hand, is a virtual implementation of physical keyboard that allows users to type in their local language/mother tongue text directly in any application without the work of direct copying and pasting. It works on a method based on dictionary-based phonetic transliteration approach, which means that whatever you type in English characters, it matches the characters with its predefined dictionary and transliterates them. It also gives suggestions based on the predefined dictionary for matching words.

One of the examples of this kind of app is Google IME. Google IME is a set of typing tools provided online by Google for top used languages, including Amharic, Nepali, Chinese, Greek, Gujarati, Sanskrit, Hindi, Punjabi, Kannada, Malayalam, Marathi, Persian, Arabic, Russian, Serbian, Tamil, Telugu, Tigrinya, Japanese, Bengali, and Urdu. Google transliteration (formerly Google Indic Transliteration) is a transliteration typing tool for Hindi and other Indic languages. This tool first published in Blogger, which is Google's popular blogging service. Later on, it was published as a separate online feature. Its popularity got it implemented in Gmail and Orkut. In the year of December 2009, Google published its offline version named Google IME. This tool from Google is based on predefined dictionary based phonetic transliteration method. In compare to older Indic typing methods,

which work by transliterating, under a particular scheme, Google transliterates by matching the English alphabet words with a predefined dictionary. Since users do not need to memorize the transliteration scheme, the service is so user friendly that it is suitable for beginners with no experience. For transliteration among scripts, and others there was, until 2011, a separate tool provided by Google named Google Script Converter.

1.5 The Idea of Transliteration Keyboard

Translation is the method of articulating an author's intended meaning in different language in which they are not comfortable or used to. Translation tells us the meaning of words converted in another language. A transliteration does not tell us the meaning of the words, but it helps you pronounce in other language sound. Transliteration Keyboard refers to the method of mapping from one language of writing to another language based on phonetic ex changeability. With this type of keyboard, you type in Latin letters (e.g. a, b, c etc.), which are changed to characters that have exchangeable pronunciation in the other language. The main target here is that the words the translator uses do not have to be indistinguishable in both the source and target language, it is the meaning which is the essential element. Even more significantly for this discussion, the sounds of the words are commonly altogether different.

For example, in BANGLA transliteration, you can type in "amar" to get আমার which sounds like "AMAR". Transliteration endorses blurry phonetic mapping. We have to just type in your best approximation of pronunciation

in English letters and transliteration will match it with the best propositions.

When a user first introduced to computer input system he/she learns about one of these input layouts. Most people are accustomed to QWERTY based layout. All of these input layouts are English language based. But there are only 360–400 million native English users. What about those users who are used to these layouts because of daily computer use and wants to write/communicate in their mother language? Here is the main reason behind developing a transliterated based keyboard.

1.6 Why Transliterated Keyboard

Normal frequently used computer keyboards are contrived to send scan-codes to the operating system, rather than immediately sending characters. After that, the stream of scan-codes is changed over a character stream by keyboard layout control application. This allows an abstract keyboard to be actively mapped to any number of layouts without exchanging hardware ingredients, just by changing the software that represents the keystrokes. It is usually possible for an advanced user to change keyboard function, and third-party applications are usable to modify or extend keyboard operations.There are different types of keyboard layouts available for interacting with personal computers. For example, QWERTY, AZERTY, QERTZ, QZERTY etc.
QWERY is by far the most widespread layout in use, and the only one that is not confined to a particular geographical area.

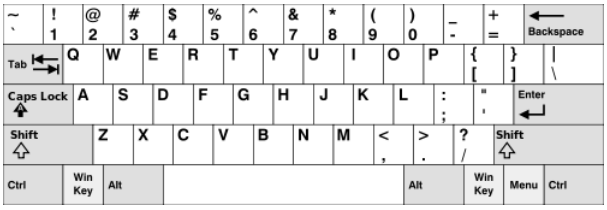

Figure 1.3: Layout 1

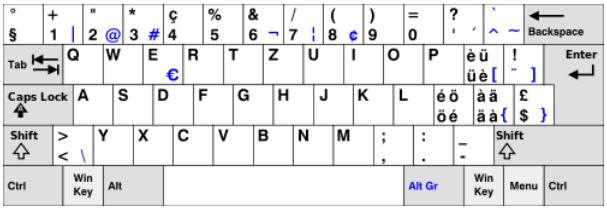

Figure 1.4: Layout 2

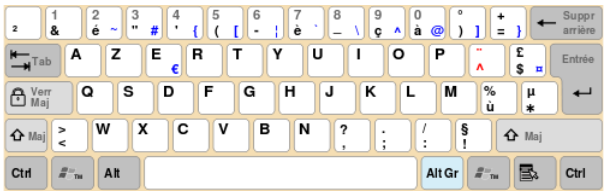

Figure 1.5: Layout 3

Figure 1.6: Layout 4

Chapter 2

Previous Works

There are not much works are in done in desktop environment while it comes to phonetic keyboard. But mobile platform like Android markets thrives with this kind of typing tools. Here are some notable mentions for desktop environments,

Akkhor is Bangla based application, developed by Khan Md. Anwar's Salam, was first published on 1 January 2003 for free. The Unicode/ANSI-based Akkhor Keyboard is usable with pre-defined keyboard layouts, letting in the Bijoy keyboard layout. Akkhor also allows a customization service for designing pre-defined keyboard layouts. It provides a keyboard handler which works system wide and also allows an autonomous Akkhor Word processor.

Bangla-onkur, developed by S. M. Raiyan Kabir, was first published on 30 March 2011 as an open source application. It eases only phonetic typing in Macintosh (MAC OS) platform. Bangla-onkur phonetic provides a user to write Bengali by typing the phonetic constitution of the words in English language keyboards. This was the commencement of phonetic input method

development.

Aparajeyo Bangla Express is Bangla Input system. This is this the initial Bangla OCR application for Bangla Language. Aparajeyo Bangla Express published other programs based on Bangla. Among them their dictionary, on screen Unicode keyboard, keyboard layout creator is worth mentioning.

Saon is a library which allows the Saon Bengali input system for touch based typing in Bengali on Linux operating systems and the project was registered by its creator, Saoni at SourceForge.net on 2012. This free and open source IM is Unicode conformable in terms of both standardization and number of keystrokes used to input a single letter. Saon Bengali provides touch typing. It is also phonetic and has something in common with all Bengali phonetic layouts making the conversion smooth for total beginners.

Avro Keyboard , built up by Mehdi Hasan Khan, was first published on 26 March 2003 for free. It provides both predefined and phonetic layouts. Avro phonetic allows a user to write Bengali by typing the phonetic constitution of the words in English language keyboards. Avro is usable as a native IME on Microsoft Windows, Mac OS X and Linux operating systems.

There is also a free transliteration web-site and software application for Bengali scripts from google.

Chapter 3

Methodology

3.1 Input

To develop a transliterated desktop based keyboard, we had to first consider the way of capturing keyboard inputs from computer for the purpose of converts these inputs into Bangla. We found that capturing keyboard input can be done by calling the API of system Dynamic Link library files.

A hook is a mechanism or technique by which a piece of code or application can intercept events like messages, mouse movement or actions, and keyboard strokes. A function that implements hook functionality to intercept events is known as a hook procedure. A hook procedure can be triggered on each event it captures and then modify or discard the event. There are many types of hook supported by windows system such as WH-CALLWNDPROC , WH-CALLWNDPROCRET ,WH-CBT , WHDEBUG etc. Each type of hook is set for monitoring different aspect of system event and message handling techniques.

We used WH-KEYBOARD-LL for our purpose. This hook hook provides information for monitoring keyboard input events about to be delivered in a thread input queue.

3.2 System Library

To implement a low level global keyboard monitoring system we needed to call windows Dynamic Link Library user32.dll's API. User32.dll is the source of most common windows API's that provides us interaction with the operating system at lower level.

We used some methods from this API described below:
SetWindowsHookExA : Set-ups an application-defined hook mechanism into a hook chain. This method install a hook procedure to monitor the system for certain types of events or messages. These events are associated either with a specific thread or with all threads in the same operating system as the calling thread.
LowLevelKeyboardProc : An application-defined or library-defined callback function used with the SetWindowsHookEx function. The method calls this function every time a new keyboard input event is about to be posted into a thread input queue.
CallNextHookEx : Passes the hook information to the next hook mechanism in the current hook chain. A hook mechanism can call this function either before or after processing the hook information.
UnhookWindowsHookEx : Removes a hook mechanism installed in a hook chain by the SetWindowsHookEx function.

3.3 System Library

From our developed code till now we get two types of output, KeyDown event and keyUp event. Means when a keyboard key is pressed, we get a message when user pushes a key down and another message when user releases a key. From this two messages we have bulit some new events.

1. KeyDown message and KeyUp message forms keyPress events.

2. Shift + KeyDown message elivers upper letter KeyPress events.

3. Digit extraction event [same as letter 1]. Now that we have successfully extracted Upper case letters ,a lower case letters and Digits , our input from user is completely ready.

3.4 Bengali Parser

Parsing, syntax analysis or syntactic analysis is the process of analyzing a dataset or string of symbols, either in human used languages, computer languages or data structures, and following to the rules of a pre-defined grammar.

Parser is the main engine of this transliterated keyboard. Parser takes English words/sentence that sound like Bangla into an actual Bangla words/sentence. For example if a user types “bangla” in traditional layouts the parser uses this as input and returns “বাংলা” as output. Here is an example table of key mapping.

	Output	Key	Output
a	আ	n	ন
b	ব	o	অ
bh, v	ভ	s	স
c	চ	p	প
ch	ছ	f	ফ
d	দ	q	ক
dh	ধ	sh	শ
D	ড	jh	ঝ
g, G	গ	N	ণ
gh, Gh	ঘ	Rh	ঢ়
j,J	জ	Sh	ষ
l	ল	R	ড়
e	এ,	r	র়
f	ফ	U	ঊ
z	য	t	ত
m	ম	th	থ
g	গ্, ঘ	T	ট
tt	ত্ত	Th	ঠ
h	হ	u	উ
i	ই	v	ভ
j	জ্, ঝ	w	ও
k	ক্, খ	x	জ

Figure 3.1: Key-mapping Example 1

For every pattern the parser tries to match the rules which are implemented for that pattern. And replace the letters by following the rules. But this rules doesn't always come in handy in case of compound letters. Because the parser cant not interpret them properly. A different rule had to be introduced to solve this problem. We assumed that If there is no vowel between two consonants it considers it as a compound letter. For example, word "rokto" will be converted to " " by our parser. It will consider" kt" as " " because there is no vowel between them. Here are some example,

Key	Output	Key	Output
Sc	শ্	St	স্ত
Sn	শ্ন	Sm	শ্ম
oZ	অ	TT	ট্ট
Tm	ষ্ম	tn	ত্ন
tm	ষ্ম	aZ, AZ	অ্যা
x	ক্স	qq	ঁ
bhl	ব্ল	bdh	ব্ধ
bj	জ্ঞ	bd	ব্দ
bb	বব্	bl	ব্ল
vl	ভ্ল	cNG	চ্ঞ
cch	চ্ছ	cc	চ্চ
dhn	ধ্ন	dhm	ধ্ম
dgh	দ্ঘ	ddh	দ্ধ
db	দব্	dv	দ্ভ
dm	দ্ম	DD	ড্ড
dg	দ্গ	dd	দ্দ
dhn	ঘ্ন	Ghn	ঘ্ন
gdh	গ্ধ	Gdh	গ্ধ
gN,GN	ণ	gn,GN	গ্ন
gm, Gm	গ্ম	gl, Gl	গ্ল
gg, GG,gG,Gg	জ্ঞ	hn	হ্ন
hn	হ্ন	hm	হ্ম

Figure 3.2: Key-mapping Example 2

Example Code: Parser is implemented in called parser.cs , wherer to-bangla(String S) is method which converts english inputs into actual bangla.

```
parser parsertest = new parser();
String bangla = parser.tobangla("amar sOnar bangla");
Debug.Writeline(bangla);
```

Output : আমার সোনার বাংলা

3.5 output

Now our goal was to deliver the parser produced output to the application that user was typing. When the user is typing, the typed letters or digits and the compiled Bangla words with suggestion is show in a dynamic box positioned near caret. This box moves with the caret position. After writing a word users generally press space or enter button. When the space or enter button is pressed compiled Bangla word is displayed on targeted writing field.

Note that this keyboard does not provide real time output display. We at first tried to show real time output display but this makes the system slow and creates complication in code. So, we decided to display the output when user finishes writing a word and presses enter or space.

3.6 Bilingual Switching

This keyboard supports both Bangla and English writing mode (two-way mode). Modes can be switched. We have introduced a hotkey for switching(F10). In Bangla typing mode user can type Bangla with suggestion and prediction and same goes for the English typing mode.

3.7 Word Suggestion

We have provided word suggestion feature in which this keyboard predicts the rest of a word a user is typing. In our designed graphical user interface user can select to accept a suggestion. This feature speeds up user interac-

tion when it correctly predicts the words a user intends enter after only a few characters have been typed.

We have used trie/prefix tree data structure for implementation of this feature. Threre are other ways creatind dictionaries for word suggestion but we chose trie because,

1. In searching for a word in a big paragraph, other pattern search algorithm like KMP algorithm takes complexity of O(n+k) where n is length of paragraph, and k is length of word we are searching. In this trie can be constructed and searching time can be reduced up to O(k) i.e. length of word.
2. There are no collision of keys in trie data structure.
3. Trie structure can provide alphabetic ordering of the entries by key.

When the keyboard starts it creates a dictionary from provided data. It takes about 00:00:01.1624092 time to train 939005 words. Notable methods in this implementation are described below,

void AddWord (string word)	This method adds words in trie.
int RemoveWord (string word)	This method prints saved words in trie.
IEnumerable<string>GetWords (string prefix)	Returns all words with specific prefix.
int Count()	Counts total words in the trie.
void Clear()	Clears trie.

Table 3.1: Notable methods

Only GetWords (string prefix) is used for collecting suggestion and other methods are for debugging purposes.

When a user starts typing characters, a depth-first-search was performed to find out what words are matching according to the typed characters. The returned words are shown in a graphical user interface and it replaces or auto-completes current the word that was being typed.

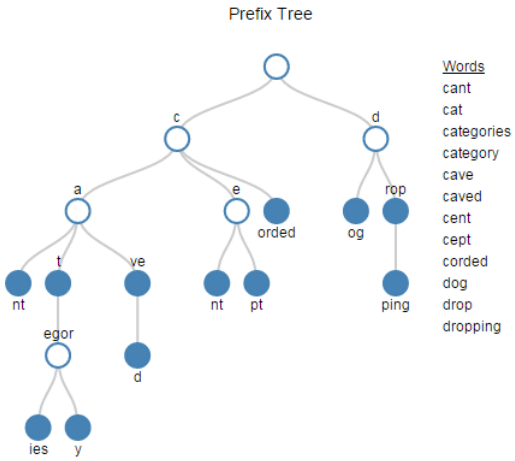

Figure 3.3: Prefix tree

3.8 Word Prediction

Word prediction is one of the important features in our keyboard. Currently word prediction is provided only in keyboard software like Gboard-the Google keyboard, Ridmik keyboard, Swiftkey etc. Where all of them are android platform based. But there is no keyboard software for desktop platform which can provide word prediction feature. Ekushe keyboard for desktop platform provides word prediction and it is provided for Bengali language.

3.8.1 Data

Our main purpose is to provide word prediction of this software for daily use. So, need a dataset which covers all the common word sequence. On the other hand, we have to keep the size of our dataset as small as possible, because the sequences of words are always kept in memory and use it in real-time. So, we need a quality dataset to provide optimal word prediction. We have tried different dataset to train our model, then we have seen that dataset from Pipilika(Document categorization data) provide best result and the size of the dataset is not very large.

3.8.2 Data preprocessing

Initially the data was not ready to be used. There are punctuations marks, unsupported character, encoding error, newline character in the dataset. We had to remove those unnecessary character from dataset. After preprocessing the resulting dataset is following:

Total Number of unique sentences	15000
Total size of the dataset file	10 Megabyte

Table 3.2: Data information

3.8.3 Approaches

Long short-term memory (LSTM): LSTM is one of the renowned approaches for word prediction. LSTM predicts next word based on a sequence of words. We have used Python keras model for implementing a LSTM model with 2 hidden layers. Our model accuracy was almost 60% and predicted next word was pretty much accurate. The next word prediction result is given below,

Given Word	Predicted Words
আমরা	জানি, এই, বলছি, সবাই, কিছু
আজ	পহেলা, দুই, যাঁরা, হরতাল, তৈরি
একুশে	বাংলা, ফেব্রুয়ারি, ভেতর, কাছে, তীব্র
বাংলা	কিবোর্ডের, ভাষায়, আগে, যাদের, বন্যার

Figure 3.4: Prediction comparison LSTM

Some problems arise when we want to use LSTM model for word prediction. To run a trained LSTM model, it is necessary that TensorFlow and some other libraries are installed in the user's system. If we provide the necessary, package in the installer file then the file size would be more than 70 megabytes and the required configuration of the user's system must be high. There was another major problem that, it takes minimum 1-3 seconds to predict next word. Which is not much effective for real-time word prediction. So, we must discard this approach.

3.8.4 N-grams

N-gram is a sequence of n words. We have implemented both 2-gram for word prediction. To calculate 2-gram we have counted frequency of a word after each word. Then we store the resulting frequency in memory and predict next word based on this frequency. Using this approach we have found that it is much efficient for real-time word prediction and its time complexity is for worst case is O(n) and for average case O(1), where n is the total number of unique words. And the result of the 2-gram is given below,

Given Word	Predicted Words
আমরা	জানি, অর্জন , আশা , এই ,একটা
আজ	দুই, পহেলা ,আজকালকার
একুশে	ফেব্রুয়ারি, বাংলা ,একুশ , একাত্তরে
বাংলা	কিবোর্ডের, ,ভাষায় , বাংলাদেশ ,বাংলাদেশের, বাংলাদেশে

Figure 3.5: Prediction comparison N-grams

We can see that the result is almost same as the result from LSTM. The main advantage of using this 2-gram model is that, it does not require any external library, which reduces the installer package size. It also required less computational power than the LSTM model. The time to generate next word in this model is less than 1 second. This is why we choose this model for next word prediction.

3.9 System Diagram

Here we represent an overall working diagram and data flow of the system.

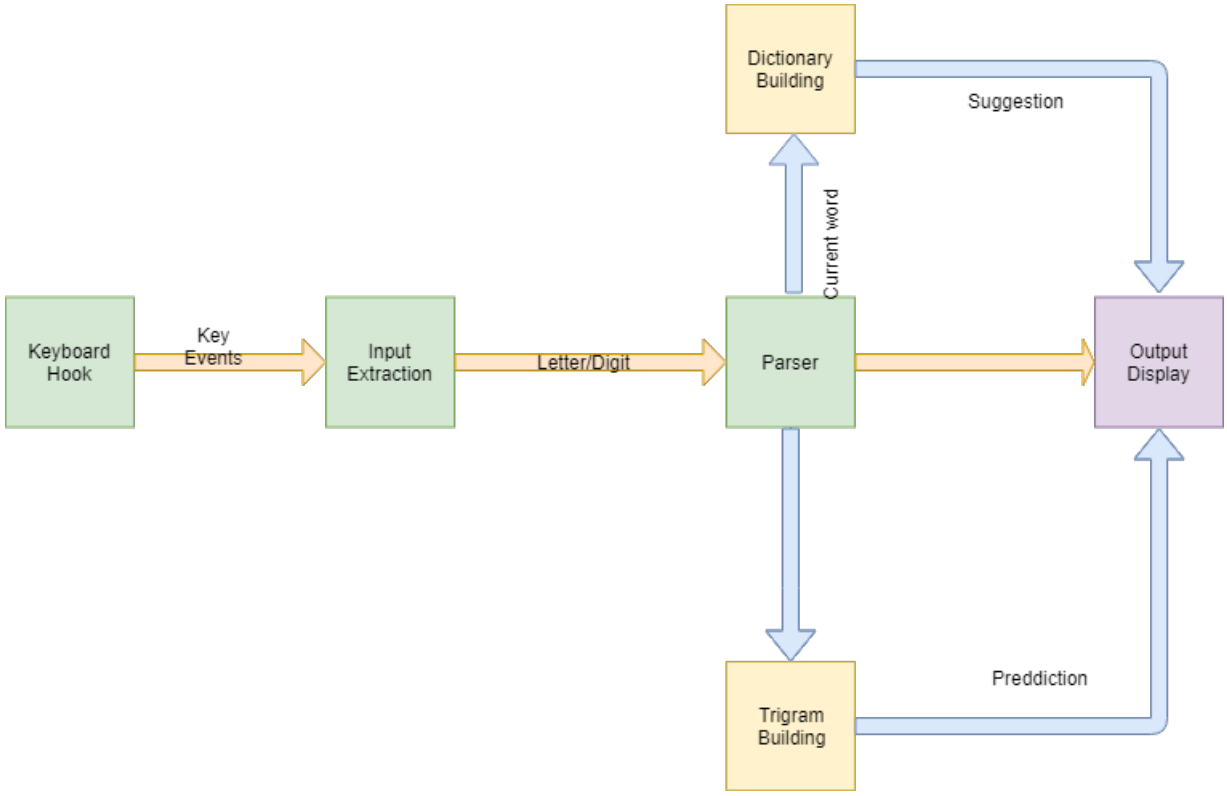

Figure 3.6: System diagram

Chapter 4

System design

In this chapter we will discuss the current state of the user interface of our keyboard software and we will show some graphical demonstration.

There are two GUI parts in this application. One is control window that floats top right corner of the display when keyboard application is running. This part contains mode switching and other necessary settings to control the application.

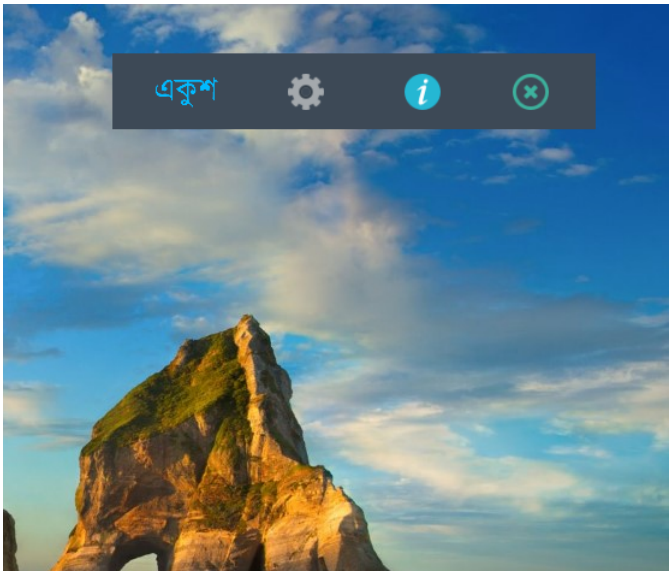

Figure 4.1: GUI sample 1

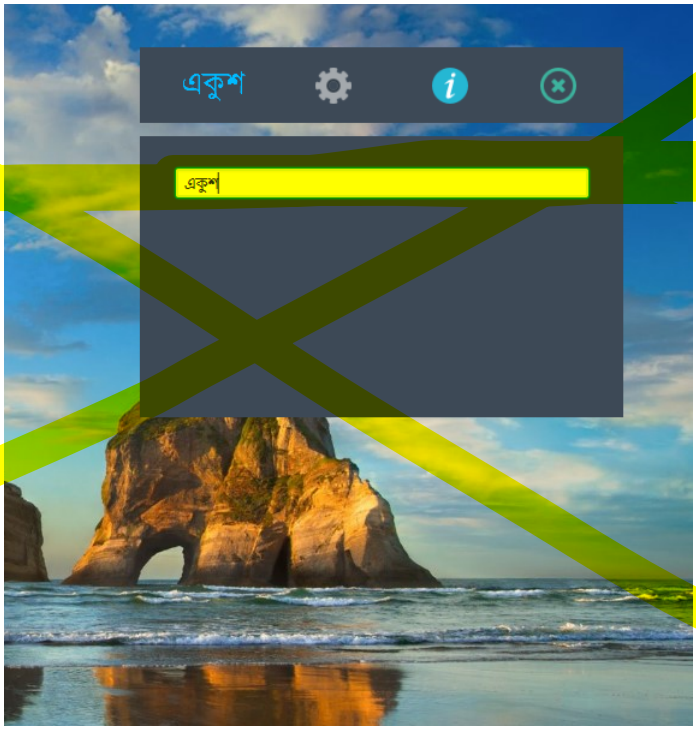

Figure 4.2: GUI sample 2

The second part of the GUI is complex box where suggestion, prediction, current typed word and current compiled word is shown. When user is typing a total of five suggestion is displayed in the box and when the word fully typed a total of five prediction is displayed in the box.

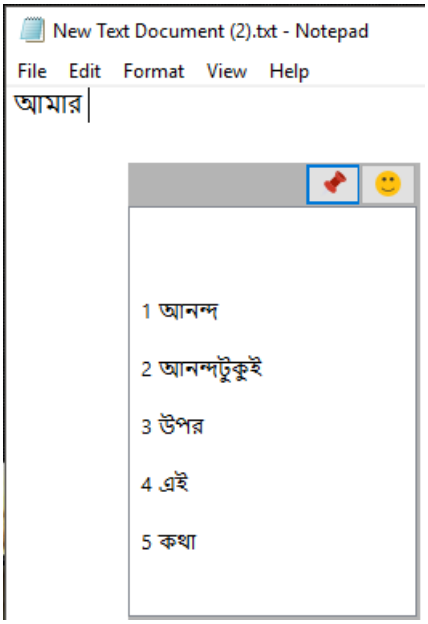

Figure 4.3: GUI sample 3

Position of this box is dynamic. When the user is typing, this box moves with the caret position. So we also implemented a button that can be used for adjusting it in a specific position.

For faster typing we implemented selection of suggestion and prediction with number from the list. If the suggestion and prediction accuracy is good enough the user don't even have to type. There is also an emoji button from where user can select up to 25 different emoji.

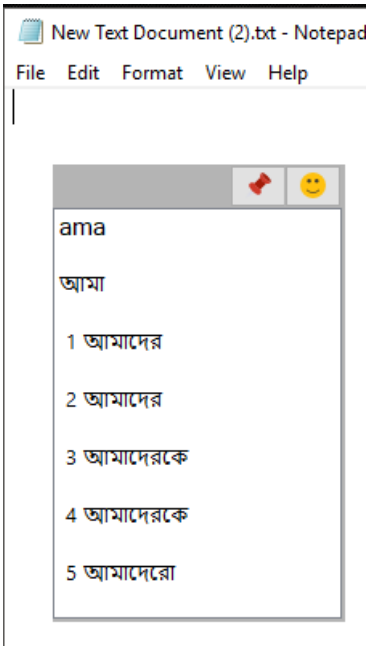

Figure 4.4: GUI sample 4

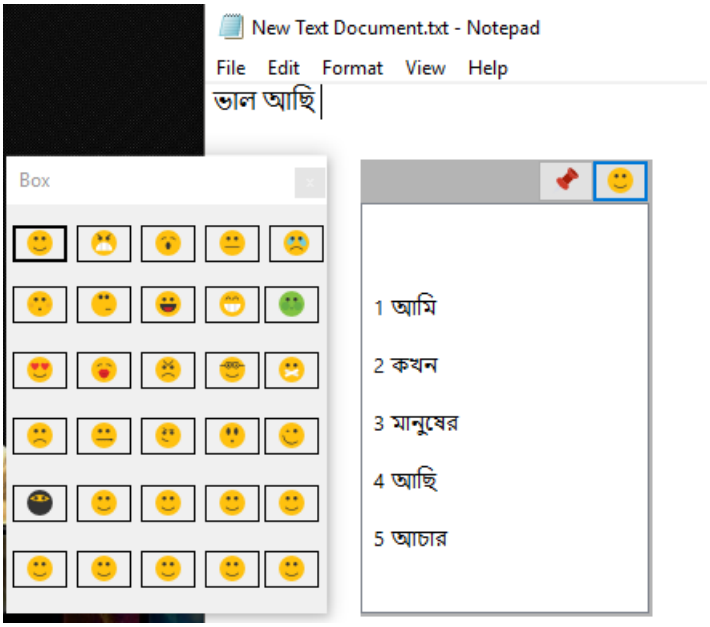

Figure 4.5: GUI sample 5

Chapter 5

Data Collections

We have collected data from our seniors of 2013 batch Gautam Chowdhury and Buddha Chandra Banik because of the similarity of data structure and training method required for suggestion showing they used in their project. They have collected data from different social media posts and celebrity pages.

They already had collected 5 lac sentences from more than 50 public pages of popular celebrities, newspapers and political groups, GitHub open source data corpus. They also have created a list of 25000 English words which is most frequently used based on the frequency of these words in the data corpus and added these 25000 words in our system so that suggestion of word works better.

Chapter 6

Problems

Our developed system is not perfect. There are some problems that we could not solve. They are pointed below.

1. Long pressing Backspace button makes the system slow.

2. Some apps (non-windows apps) does not provide process ID. In this case, dynamic positioning of second part of GUI does not work and user must adjust GUI position manually.

3. As we implemented suggestion and prediction selection with numbers, when the user want to type digit after typing word , it creates an ambiguity about selecting next word and digit typing.

4. Some rare performance issue while typing very fast.

Chapter 7

Conclusion

This report represents entire system structure, method, design, evolution till now since we started developing of this keyboard. The current condition of the keyboard is, user can type Bangla and English perfectly with it and can get word suggestion and prediction based on what they are typing. It has a graphical user interface to control settings and switching modes.

Hope this keyboard will help targeted user with proper and fast Bangla/English typing. We hope to get positive feedback from our user's.