Design of Predictive text input method for Swarachakra Marathi

Semester III Project

Guided By
Prof. Anirudha Joshi

Prasad Ghone
Interaction Design
M.Des (2014-2016)

Industrial Design Centre
Indian Institute of Technology Bombay

Semester III Project Report

Prasad Ghone,
146330010,
Interaction Design, Master of Design

Guided by Prof. Anirudha Joshi

Industrial Design Centre,
Indian Institute of Technology Bombay

Declaration

I declare that this written document represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources.
I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission.
I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Prasad Ghone,
146330010

Interaction Design
Industrial Design Centre,
Indian Institute of Technology Bombay
November 17, 2015

Approval Sheet

The project titled *Design of Predictive text input method for Swarachakra Marathi* by Prasad Ghone, is approved for partial fulfilment of the requirement for the degree of 'Master of Design' in Interaction Design at Industrial Design Centre, IIT Bombay.

Guide:

Chairperson:

Internal Examiner:

External Examiner:

Date: 29th July 2016

Acknowledgement

I would like to express my sincere gratitude to Prof. Anirudha Joshi for his support and guidance throughout.

Thanks to Prof. Venkatesh Rajamanickam, Prof. Girish Dalvi, and Prof. Ravi Poovaiah for their valuable inputs during the course of the project.

A special thanks to Dileep Mohanan and Jayati Bandyopadhyay for taking me out of all the android development hurdles all the time. Thanks to Akshay Kore, Indrajeet Roy and Sagar Yende for all the motivation and support.

Thanks to the Swarachakra Team, IDC for all the resources and support.

Most importantly, my family for all their love, patience and encouragement.

Prasad Ghone

Index

Abstract

In the earlier experiments in our lab, it was observed that prediction mechanisms slows down the text entry speed of users[1].

This project attempts to design a mechanism for text prediction for Marathi on Swarachakra keyboard for smart phones. The 2 important hypothesized reasons which account to cognitive load were:

Interface requires users to shift attention constantly- Predictions are displayed in a prediction window which generally is on top of the keyboard. If intended prediction does not appear in the first go, user do incremental addition of characters and look for prediction. In this he/she constantly, shift his/her attention from keyboard to prediction window. This visual discontinuity is one of the reason for worse performance of current prediction systems.

Users are unable to build a conceptual of the prediction system- Novices or Intermediate users have not built the conceptual model of predictions of predictive system. For them, the behaviour of the system is completely uncertain. If there is word which a predictive system won't predict, these users still keep looking for it. This is another potential reason for worse performance of current predictive systems.

In this project, I designed a new predictive interface for Swarachakra Marathi keyboard maintaining the design continuity with keyboard. Based on the hypothesis of the long time required for conceptual model creation of predictive system and the still the uncertainty about the predictions, corpus of the new predictive keyboard is thresholded to 1000 words which covers 47.33% of Swarachakra corpus. The project is an experiment to test whether these factors have an effect on cognitive load and in-turn on the performance of the predictive systems for Marathi.

The project was evaluated with Swarachakra with prediction chakra and More corpus (Top 20,000 words which contributes to 79.88%) and Swarachakra without prediction. A within subject study was carried with 5 users. Questions like Will interface change bring a difference in performance of keyboard? Will thresholding a corpus help in fast and precise building of conceptual model of predictive system? Will prediction help at the first place in improving performance of the keyboard? were tried to answer in the evaluation. Swarachakra without prediction performed the best in all the keyboards followed by Swarachakra with less corpus.

1. Motivation

Text entry is one of the most basic thing you do on your smart phones. With the growth smart phone penetration in India[23,24] and with several emergent users[26] beginning to use Smart phones, the need of Indian language text entry has increased.

A study towards standardization of virtual keyboards for Indian languages was carried by various Indian Institute of Technology(IITs) and other institutes. In IIT Bombay, a longitudinal study with 153 users, new to virtual keyboards was carried[1]. Existing 4 virtual keyboards for Marathi on smart-phones were examined and empirically evaluated. 10 preliminary theoretical effort models were also created to simulate the effort needed for each keyboard[1]. The results were surprising. The keyboards that used prediction saved efforts theoretically, but performed poorly in the empirical evaluation.

Prediction mechanism can be considered a success only when users are using it and are able to type faster and more accurate than non-predictive keyboards. The important and fundamental questions which are raised from studies are:
Why prediction methods fail? Why is it the performance of non-predictive keyboards is better than predictive? Even for English, it is not clear whether prediction really improves typing performance[2,27]. *Even when a keyboard is expected to perform well by theory, why it performs worse in practice?* Currently, we are just using the Indian languages on the predictive systems designed for English. I*s using the same text prediction paradigm of English for Indian languages a good idea?*
There is lot of work happening around the globe for various regional languages, which takes into account the morphology of language in prediction[3]. *Will considering morphology and agglutination of the language yield better result for Indian languages?*

The three important questions which were raised are *Where to predict? What to predict? and How much to predict?*

Where to predict?
Does traditional way of predicting words on top of the keyboard deteriorate the typing performance? If the word is predicted on the keyboard itself, user will not have to shift his attention from keyboard where he is typing to the prediction bar on top. Will predicting text on the keyboard itself will improve typing performance? The project tries to explore this question of where should a prediction mechanism predict text for better results.

What to predict?
Most Indian languages are agglutinative. Marathi is an agglutinative language. Agglutinative languages have a larger vocabulary size. Lot of words are formed by suffixation. Using this complex morphology of language for prediction by predicting n-grams over complete word might yield better results. The project will try to explore the opportunities of predicting n-grams over complete words.

How much to predict?
From the expert evaluation, it was observed that, building a conceptual model of the predictive system of which all words will be predicted and which won't be predicted is a difficult and requires several typing hours. In the evaluation[1], it was

observed that the users did not pick the predictions and instead preferred typing out the word. The speculation made was the users over the time realised the cost of picking a prediction from prediction window over straight away typing out the word. Is it because the corpus is large and prediction system is trying to predict everything affecting the conceptual model making of the users. Will reducing the prediction corpus help users make conceptual model faster? These questions must to answered by evaluating them with users.

The project was started with a focus to answer all these questions. The project explored how interface of the predictive system affects its performance. Also, to see whether predicting less words improves the predictability of a system and improve performance. The future work is expected to build on top of this and consider features of Indian languages for a better predictive system.

2. Design Process

The classical design process for this project would be to conduct secondary research, primary research, generate insights and design ideas,prototype and then evaluate the best ideas. Based on the theoretical understanding of the project, evaluate ideas heuristically and theoretically. Then to finalise upon an idea based on the evaluation. This project did not follow this classical design process as the starting point of the project was different.

However, some exploratory early stage design ideas were generated for different prediction mechanisms. These ideas are described in section [Section 5].

As observed from the studies conducted in our lab, we can say, it is very difficult to speculate performance of a keyboard without empirical evaluation. In earlier studies, keyboard which was expected to perform better was proven wrong empirically.
We can generalise this to say, issues in text input especially in case of Indian languages cannot be solved without empirical evaluation. The text input method's performance can only be concluded, if it performs as expected on field.

This project started with an intent to find answers to mainly three questions described in Section 1 which are *What to predict? Where to predict?* and *How much to predict?*
Various design ideas were generated based on the combinations of the above questions. The final concepts were selected considering the academic limitations and by evaluating it based on the earlier observations. The final set of concepts were prototyped and evaluated with users. Also, the concepts which were not selected were prototyped and tested against the final concept.

3. Findings from previous studies

As discussed in the Section 1, longitudinal studies was carried for standardization of virtual keyboards. Four keyboards namely Swarachakra, CDAC inscript, Swift key and Sparsh were evaluated. The evaluation gave an opportunity to look prediction issues for Marathi text input for these keyboards.

Preliminary Effort models of these keyboards to type in a corpus was calculated theoretically. Then the data available from empirical longitudinal study conducted was compared with these theoretical data.

The corpus used for the evaluation consisted of 10 training words, 20 first time usability test (FTU) words and 300 phrases for a longitudinal evaluation (LTU). The training words were selected to represent the typing difficulties that a user typically faces while learning to type in Marathi. These training words were presented to the user in increasing order of difficulty during training described below. The twenty FTU words represent the same difficulty levels.
The LTU consists of 300 phrases, 1,421 words and 8,024 Unicode characters. The phrases were a balance of informal conversational phrases and formal phrases. The formal phrases were selected from school textbooks, popular folk songs, children's songs, classic poems, verses from the national anthem, and well-known quotes by historical personalities. The corpora phrase length ranged from 1 to 8 words with an average phrase length of 4.74 words. The phrase lengths varied from 11 to 55 Unicode characters with an average of 26.75. We classified phrases as easy, moderate and hard.[1]

InScript and Swiftkey uses government of India standard keyboard layout of physical keyboards. In this layout, many keys are shifted. To access the shifted keys user can either press shift and type the key or long press to directly type the shifted key. Flow is the gesture input mechanism by which user can type a word by swyping.

.

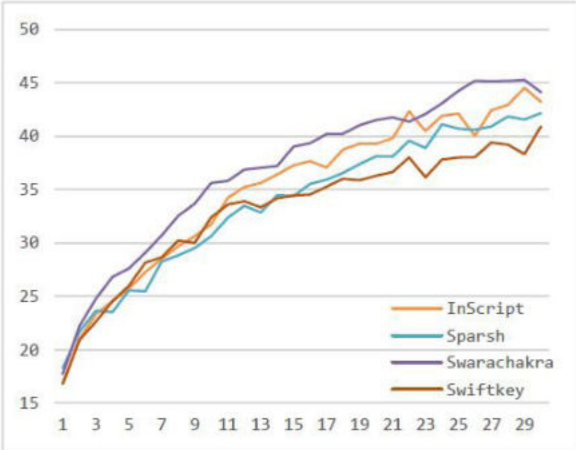


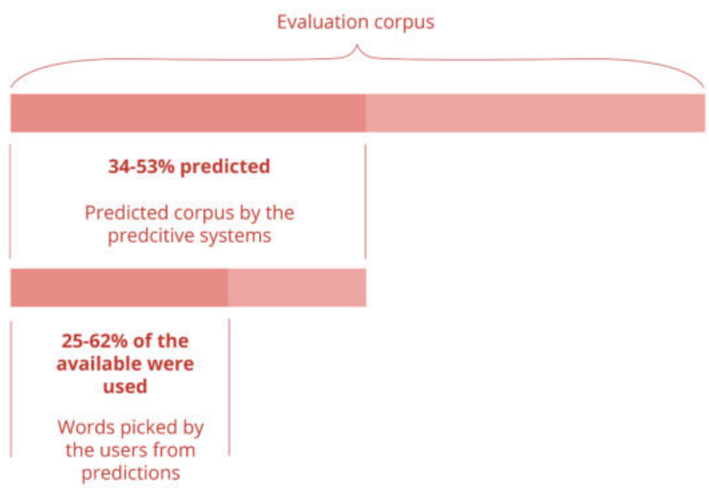
Fig. A. Summary of speeds in empirical findings

In Fig. A shows Average CPM for each keyboard in peak hours and overall.

The graph shows, the keyboards without prediction are performing better than the keyboards with prediction[1]. Swarachakra which does not have prediction is performing best even after sufficient typing practice, followed by InScript which also does not have prediction. Sparsh and Swiftkey which have prediction performs worse than the above[1].

4. Issues with prediction system

From the study, the reasons for worse performance of predictive keyboards can be explained. During Evaluation, it was observed that only 34-53% words were predicted by the predictive keyboards. Out of these predicted words users picked only 25-62% of the words. Additional cognitive load to pick the prediction from prediction window is a reason was speculated for poor performance. We call it "cognitive toll of prediction bar".



5. Hypothesis

The hypothesis was made in regard to the poor performance of the predictive systems. The speculation was made based on the empirical data from evaluation and also from the expert evaluation we underwent for standardization of virtual keyboards project.

The hypothesized reasons speculated for this poor performance were:

1. Shift of attention:

Users tend to constantly shift their attention from keyboard to prediction window to select prediction as well as type using keyboard. This visual discontinuity is one reason for worse performance.

2. Building conceptual model of the prediction system:

The users does not know which all words will be predicted by the predictive system. If there is a word which will never be predicted, user still waits for it to appear in the prediction window, which hampers the typing performance. Once the user has the conceptual model of the predictive system, he/she won't wait for the predictions which he is kind of sure won't be predicted and he/she straight away type it out. This saves user's time and reduces cognitive load.

6. Understanding Users

The User group for the project was studied and modelled. Users are classified based on their acquaintance with Keyboard, their knowledge about language and its rules and the their typing experience. Users are broadly classified into three groups:
I. Novice Users
II. Intermediate Users
III. Expert Users

I. Novice Users
Novice users are the users who hunt and pick a character. They are new to keyboard layout and the keyboard idea. They struggle with language rules of Marathi like formation of conjuncts, using rafar, etc.

II. Intermediate Users
Intermediate users know location of frequently used characters on keyboard. They know language rules of Marathi like formation of conjuncts, using rafar, etc. They can type sentences accurately at low speeds.

III. Expert Users
Knows and have a muscle memory of frequent characters. Knows language rules. They can type any word without help. They don't make mistakes at high speeds.

When prediction is available on a keyboard, this behaviour of users get affected to some extent which is mentioned below.

I. Novice Users
These are the users which struggle with keyboard itself. Using prediction for typing is not a behaviour expected from these users. These users are less open to using prediction as this adds cognitive load of using an extra interface component. This is a short lived phase. After few typing hours, these users become intermediate users. On the other hand, prediction might help these users as they don't know language rules. It can increase their accuracy and articulateness.

II. Intermediate Users
These users are active in typing as they type as well as use predictions. They have the conceptual model of keyboard and know the language rules and are building conceptual model of the predictive system. They need to have keyboard knowledge and also understand rules of text input. While using predictive system, they tend to focus more on the predictive model of the system than the actual message. These users are the target audience for the project.

III. Expert Users

After several hours of typing users become expert in typing. They type at relatively higher speeds. These users know language rules, don't make errors in typing at higher speed and have muscle memory of frequent characters. The speculation from expert evaluation is, these are the users who have built conceptual model of the keyboard and prediction system. As, they type at relatively higher speeds, shift of attention reduces their typing speed. They prefer typing the word instead of using prediction.

7. Background Study

To understand existing prediction mechanisms, current available prediction mechanisms were studied. The thought behind these explorations was to understand different prediction layouts and mechanisms. The examples shown are mostly for English or foreign language as not much novel work has happened for Indian languages. This will be followed by secondary research of the current text input mechanisms for Indian languages.

Various predictive systems are essentially trying to do three things:

1. **Word completion:** Here, words are predicted when the user is typing. User can select one of the prediction before completing the word.
2. **Word prediction:** This is basically next word prediction. Here, user has typed one word and provided that word is typed what all are the probable next words that the user might type are shown in the prediction window based on various probabilistic models.
3. **Auto correction:** Here, after user has typed a word and if that word does not exist in the corpus with relatively high frequency, that word is corrected viz. replaced by a word which the user was intending to type. The words are auto corrected based on various probabilistic models.

Probabilistic modelling of keyboard:
Prediction mechanisms are also designed based on the probabilistic modelling of keyboard. The modelling is on the keyboard level. This modelling takes care of error cases while typing, for example user pressing a key which is in proximity to the intended key, keyboard models the entire word with the error and either auto-corrects or predicts. This kind of modelling happens at word level. For example, if a user wants to type a word 'office', while typing he makes an error and types 'oddice', but then as 'd' and 'f' are next to each other, keyboard will auto-correct the word or predict the word 'office'.

Smart Gestures in prediction:

Keyboards which involve smart gesture typing use shape recognition to predict the word. Auto-correct and prediction are very important in gesture typing as the recognition is based on the probability. In these keyboards, the words are stored in the form of shapes in dictionary. The gesture which a user does or a shape which he draws is matched and the word which is most probable is selected by the keyboard. The words which also are probable with lower probability are displayed in the prediction window.

Various predictive keyboards were reviewed for the project which are shown below:

- **Existing prediction mechanisms**

The gist of prediction mechanisms are provided in the following section

1. Most of the keyboards have 3 predictions in the prediction bar. These keyboards show just 3 predictions to empower user to look at 3 distinct point for predictions rather than scanning the entire prediction bar. These kind of prediction bars usually have a drop down after the predictions for more predictions. Most of the users prefer typing the text and wait for prediction to display upfront than to look for more predictions.

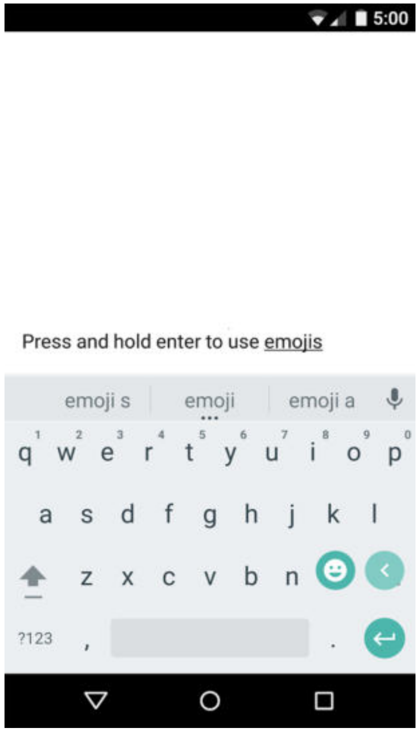


Fig. 1. Prediction Interface with 3 predictions
Source [8]

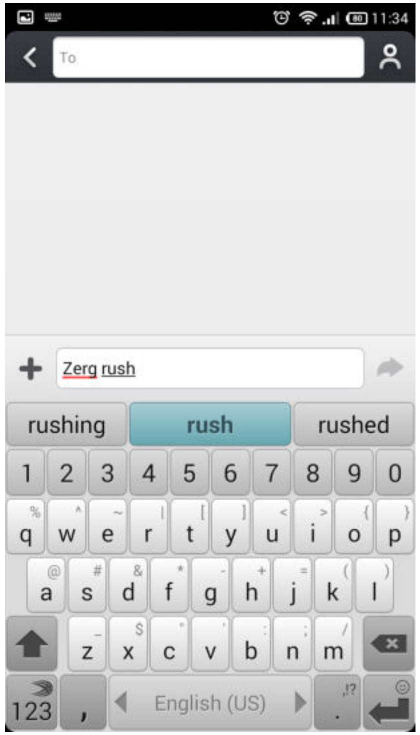


Fig. 2. Prediction Interface with 3 predictions
Source [9]

2. Some keyboards have multiple predictions than a standard of 3 predictions in the prediction bar. This increases the predictability of the words but increases the gaze time required to spot the prediction.

Fig. 3. Prediction Interface with multiple predictions
Source [10]

Fig. 4. Prediction Interface with multiple predictions
Source [11]

3. Few keyboards show the predictions right on to the key. This helps as the gaze time is reduced completely. Provided there is no prediction, user have to anyway locate that character and type it. So, prediction on the key helps in saving the gaze time. These predictions are good when user has just started to type the word. The number of possibilities of words are more. But, after few taps, if a single key may have multiple predictions. This is one of the limitation of this kind of an interface. Long words on adjacent keys cannot be predicted on these interfaces. Also, expert users do not look for next keys. The key positions are stored in their muscle memory. We speculate this kind of prediction wouldn't help much in that case.

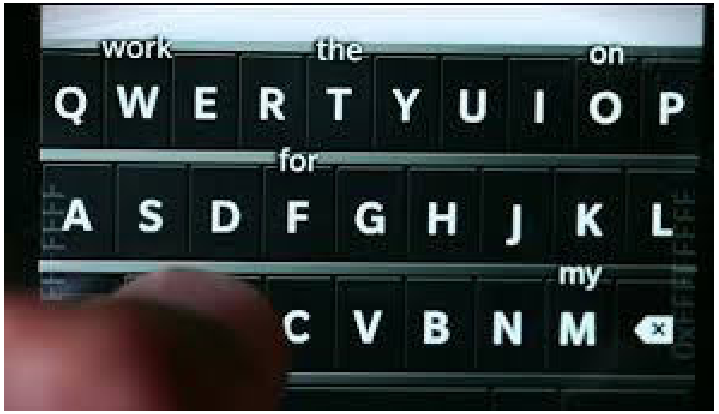


Fig. 6. Prediction Interface with prediction on key
Source [13]

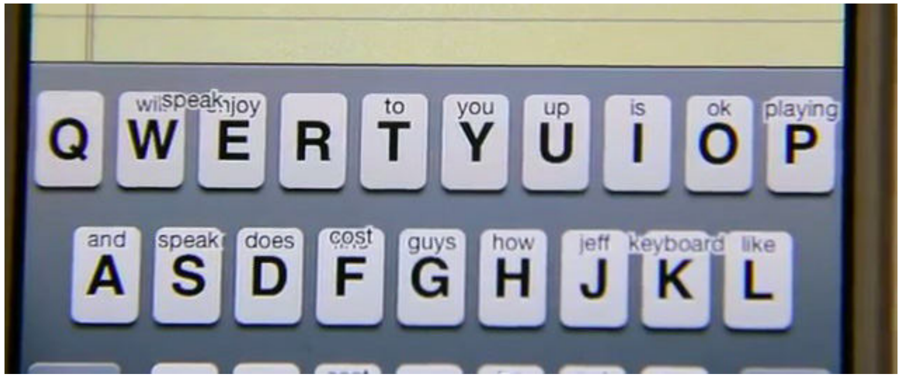


Fig. 5. Prediction Interface with prediction on key
Source [12]

Fig. 7. Prediction Interface with prediction on key
Source [14]

4. Few interfaces give an option of typing in between characters, the prediction system predicts words which have those characters. Previous studies have mentioned that prediction helps articulateness of text and spelling correction[reference], these kind of prediction mechanism helps users achieve them.

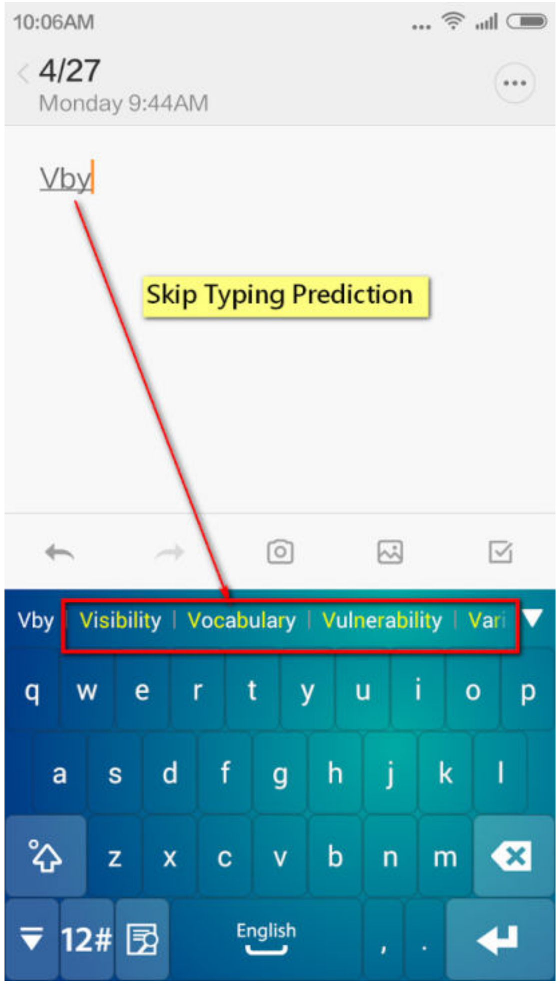


Fig. 8. Prediction Interface which predicts in between words
Source [15]

- **Experimental text input mechanisms for English for fast typing**

Few experimental text input mechanisms which claim to have faster text entry speed are presented in the section below. As by text prediction intervention, we essentially try to improve the text entry speed of users. This interfaces claim to increase the speed of typing by changing text input mechanism. It is interesting to look into how design intervention in interface improve the text input performance.

1. **8 pen** is a keyboard designed for touch devices. Here, user draws shape in the form of loop. The character which he wants to input, he draws a loop around it and the character is typed. Once, the alphabet position in the quadrants is stored in muscle memory, the keyboard claims to improve typing speed. The learning graph of the keyboard is slow.

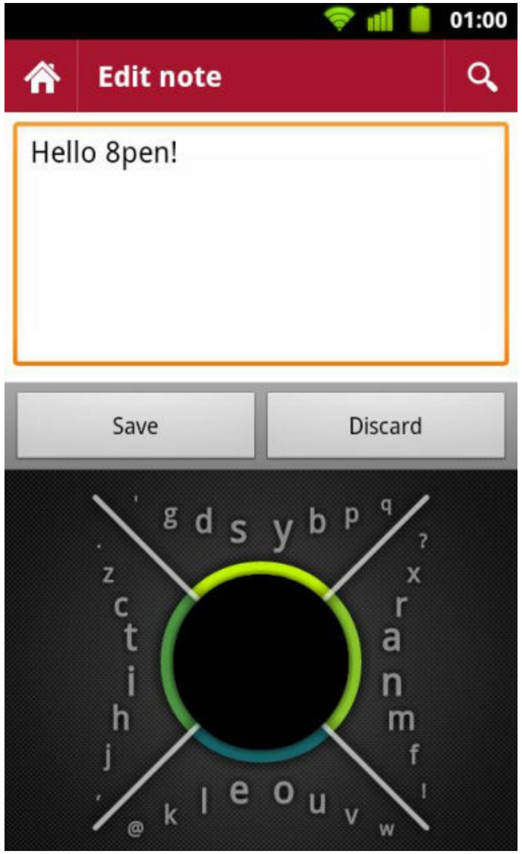


Fig. 9. Interface of 8pen keyboard
Source [16]

2. **Slice Keyboard** is keyboard designed for touch devices. The idea behind the design of the keyboard is that the user should not look at the keyboard while typing and just focus on the content and correct mistakes in typing.

Whenever, user touches the keyboard, the touch-points are set as anchor points and user can tap, drag and do gestures to type. Flicking gestures to add space, delete text is used. User can hold on multiple points at the same time. It is a simultaneous multi touch keyboard.

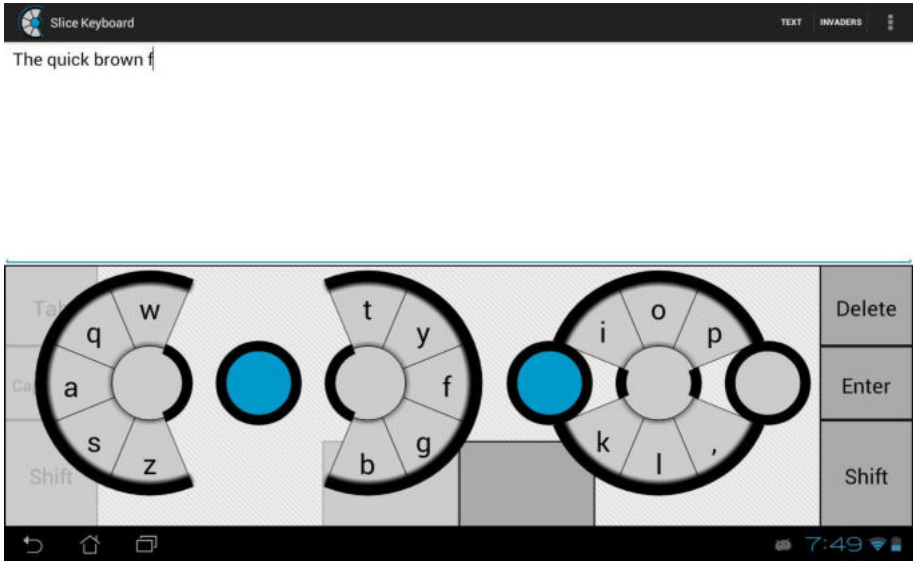


Fig. 10. Interface of Slice Keyboard
Source [17]

3. **Dasher keyboard** is a keyboard which is made for devices with a pointer. To use this keyboard for text entry, user doesn't need a keyboard. User use a pointer device by which he can zoom to see the letters and then select one of them. It also has an prediction feature. Letters are pushed based on the probabilistic model for prediction.

The important advantage of this keyboard is that there are no two different modes of typing and prediction selection. User selects text while he is typing[25].

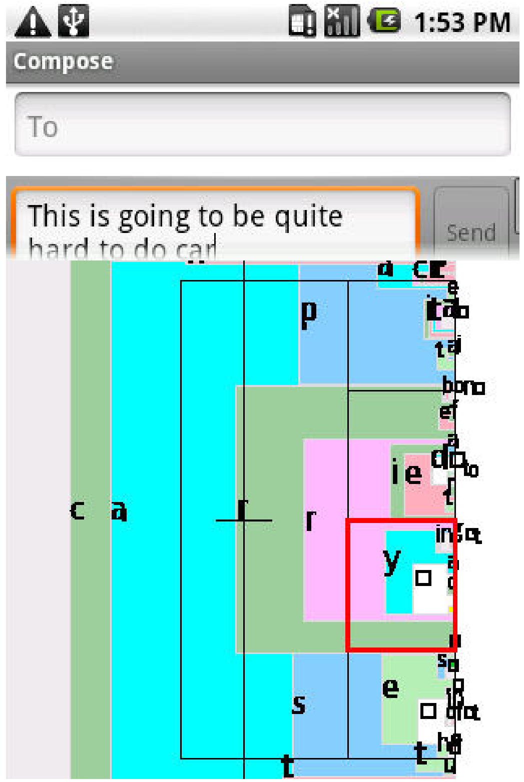


Fig. 11. Interface of Dasher Keyboard
Source [18]

3. **MessagEase keyboard** is a keyboard which is made for faster typing. It says that it improves your typing speed when you are acquainted with keyboard[26]. In this keyboard, there are 9 huge keys which are the most frequent used letters, each big has 8 directions to type in. User can either just tap and write the text. Or he can either do gestures and draw shapes to type. He can also used both approaches.

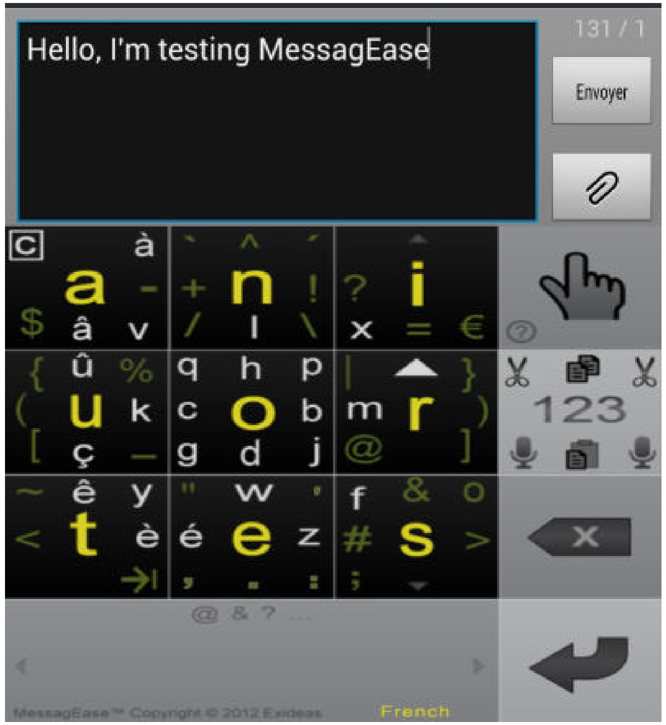


Fig. 12. Interface of MessagEase Keyboard
Source [19]

- **Indian language text input mechanisms**

For Marathi, there are several keyboards available for smart phones on app stores. The two main categories of keyboards are logically arranged keyboards and frequency based keyboards. The logically arranged keyboards are the keyboards, in which keys are arranged on the basis of logical structure of language and script. This the way the script is taught in schools. Whereas, frequency based keyboards are the keyboards in which the keys are arranged on the basis of frequency of use. The keys which are frequently used are placed on low effort positions.

These are the few keyboards mostly used by users for marathi text entry on smart phones:

1. **Sparsh** is a logical keyboard. It has prediction feature in it. It predicts the word user is currently typing. The prediction is displayed in the prediction window which is above the keyboard. The number of predictions are not fixed and depends on the number of predictions the prediction window can accommodate. It used frequency based prediction mechanism.

Fig. 13. Interface of Sparsh Marathi Keyboard Source [20]

2. **Swift key** is another most used keyboards for Marathi. It is also a logical keyboard. It has features like word completion and next word predictions. User can either type or make flow gestures to type. The prediction window is on top of the keyboard, with three predictions. It uses n-gram prediction mechanism.

- The prediction here is non-contextual. It does not understand the context of the text. For example, when a user wants to type a phrase तळे राखी तो पाणी चाखी , As soon as user types राखी, the next word which is predicted is सावंत because the frequency of सावंत after राखी is higher than frequency of तो after राखी.
- By swiping, users are held to keyboard and are not dodging between keyboard and prediction window till the word is complete. This reduces the shift of attention problem, but if the word is not predicted the effort is wasted. So, it is expected that the user must have understanding of the words keyboard predicts.

Fig. 14. Interface of Swift Key Marathi Keyboard

3. **CDAC Inscript** is another most used keyboards for Marathi. It is a partial frequency based and partial logical keyboard. The position of the keys is decided based on the frequency of use. It is the keyboard layout which is standardized by Indian government for physical keyboards. It has prediction feature in it.

As all the vowel modifiers are exposed out, most of the consonants are hidden and can only be seen in shift mode or after long press.

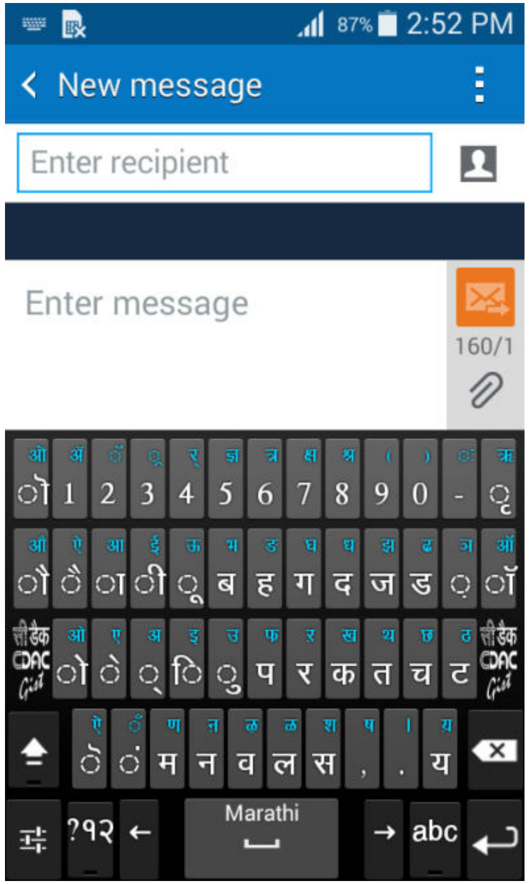


Fig. 15. Interface of CDAC Gist Marathi Keyboard

4. **Swarachakra** is a keyboard designed by IDC, IIT Bombay. Swarachakra Marathi was found to be the best in the comparative studies of keyboards for Marathi in which Swarachakra was evaluated with inScript, Swift key and Sparsh keyboards. Swarachakra is a logical keyboard and the keys are positioned as per the language script, the way the script is taught in schools. It contains a pop up UI element to display vowel modifiers. It is a non-predictive keyboard.

When a user taps on a consonant, all vowel modifiers pop up in the form of a chakra as shown in the adjoining figure. User then drags his/her finger to select a vowel modifier.

Fig. 16. Interface of Swarachakra Marathi Keyboard

8 . Understanding Problem

The Problems involved in current predictive interfaces are: (These are based on the observations of expert evaluation of keyboards. As discussed earlier, these are the reasons we speculate, answers to which we will get while evaluating the solution)

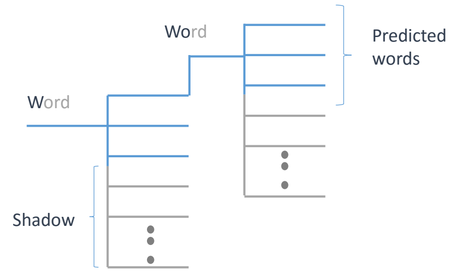
Shift of attention: User has to constantly shift his attention from keyboard to the window where prediction is appearing. This visual discontinuity hampers the performance of the predictive system.

But one might definitely ask why is it that prediction did not appear at the first place? Why user has to do incremental changes and check if the word is predicted or not. One of the potential reason for this which we speculate is this can be due to large shadow size.

A shadow are the words which are not getting predicted because of prediction window limit. Most of the Indian languages are agglutinative in nature (Marathi being is subject, it is agglutinative) i.e. various words can be formed by inflection. So, the words which get an opportunity to be displayed in the prediction bar are not significantly high probable than the desired word not predicted.

The above figure shows the predictions appearing when user has typed आ, the first three words are in the prediction window, whereas the entire corpus starting with आ except first three are in shadow.

The data from the actual corpus was analysed to see the results. Table below shows the prediction and shadow information when a user has typed आ.



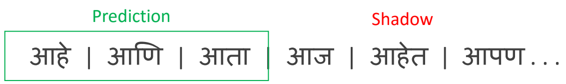


Fig. 22. Concept of prediction shadow

The adjoining figure shows the information of predictions and shadow when a user has typed आ. The highlighted words show the words that are predicted and get the privilege to be displayed in the prediction window, whereas the subsequent 3 words shown lie in shadow. Shadow consist of all other words which are not displayed in the prediction window. These words and their frequencies are used from the corpus [Note 1] used in the study. The frequency of the prediction window and the shadow are also shown in the adjoining figure. Percentage of corpus covered in the prediction window and percentage of corpus in shadow is highlighted.

The next three figures also shows similar data for the input words ना, का and ए respectively.

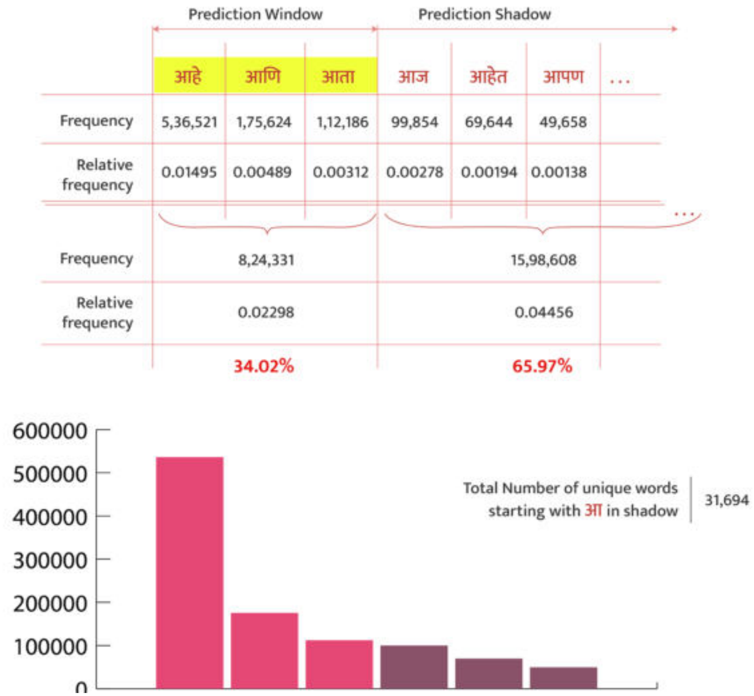


Fig. 23. Prediction and shadow information if आ is typed

Note 1: Here the corpus is Swarachakra corpus, which can be found here[4]. The corpus contains 3,58,71,284 words with 17,34,282 unique words.

	Prediction Window			Prediction Shadow			
	नाही	नाय	नाव	नाहीत	नाम	नाहीतर	...
Frequency	3,37,670	25,314	25,077	14,493	10,856	7,544	
Relative frequency	0.00941	0.00070	0.00069	0.00040	0.00030	0.00021	
Frequency	3,88,061			2,26,193			
Relative frequency	0.01081			0.00630			
	63.17%			36.82%			

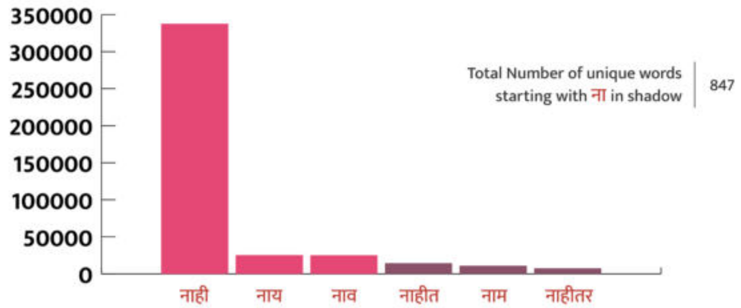


Fig. 24. Prediction and shadow information if ना is typed

	Prediction Window			Prediction Shadow			
	काय	काही	काम	कारण	काल	काळजी	...
Frequency	3,34,091	76,197	50,204	30,095	20,090	10,811	
Relative frequency	0.00931	0.00212	0.00139	0.00083	0.00056	0.00030	
Frequency	4,60,492			3,88,418			
Relative frequency	0.01283			0.01082			
	54.24%			45.75%			

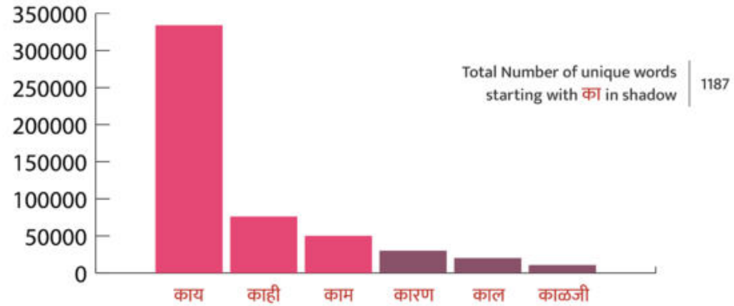


Fig. 25. Prediction and shadow information if का is typed

	Prediction Window			Prediction Shadow			
	एक	एका	एकदा	एकदम	एकच	एकत्र	...
Frequency	10,54,65	22,490	12,800	11,294	10,205	3,693	
Relative frequency	0.00294	0.00062	0.00035	0.00031	0.00028	0.00010	
Frequency	1,40,755			1,25,627			
Relative frequency	0.00392			0.00350			
	52.83%			47.16%			

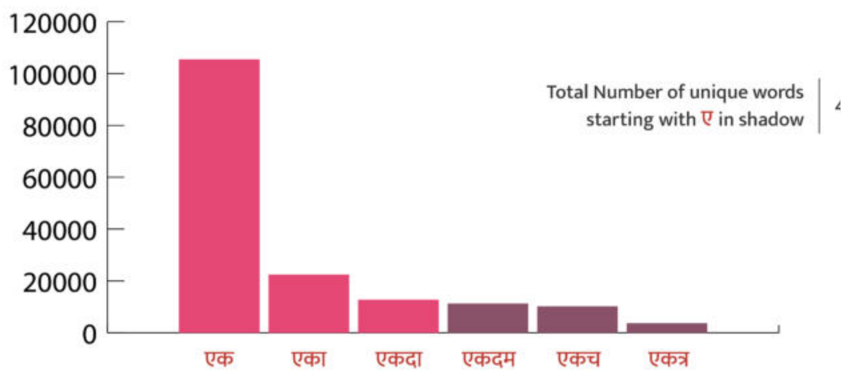


Fig. 26. Prediction and shadow information if ए is typed

It can be seen that the words in the prediction window are not significantly high than the shadow. User typing a word from the shadow and not from the prediction window is highly probable. So, we speculate, n-gram prediction will make prediction window more significant and predictability of the system will go high.

However, in the current scope of the project, using concept of prediction shadow for better predictability is dropped.

Building conceptual model of the prediction system:

The user does not know which words will be predicted by the prediction system. If there is a word which has never been predicted by the system and will never be predicted, user still looks for the word in the prediction window as it seems to be frequent enough. Over the time he builds a conceptual model of words which are predicted and words which are not. Then after, he won’t look into the prediction window to check prediction of the word which he doesn’t expect to be. This is an expert behaviour and comes after several typing hours. For example, in the example below, a user who is new to prediction will look for all the predictions if he had to type these both statements. But, over a period, an expert user won’t look for “मंगलदायक” in prediction window. He will straight away type it. Even if that was predicted. Because, till then user has an understanding of cost of the looking for a prediction in an uncertain space to actual effort needed to type without prediction.

औषध घेतले दुखणे थांबले

जन गण मंगलदायक जय हे

9. Redefined Brief

The project aims to build a predictive interface method for Swarachakra Marathi. After understanding various linguistics, algorithmic and interface problems of a prediction mechanism for an Indian language on Smart phones. A novel interface is created to solve specific problems a user faces. The problems to solve and test for these project are:

1. Shift of attention problem.
2. Longer time taking and still uncertain conceptual model of predictive mechanisms.

The target audience for this project will be mainly intermediate and also expert users described in above section.

10. Initial Design Ideas

1.

Bringing the standard prediction window from top of the keyboard to chakra.

In this idea, the aim was to minimise the shift of attention problem. The prediction bar from the traditional top position was brought to position just above chakra in Swarachakra. User generally tend to type without looking at the editor box. This idea, helps user do that. He doesn't have to look at the editor box and select predictions right near the characters.

Limitations:

1. Selection of predicted words:
Coming out of the chakra and selecting predictions is not intuitive.
2. For Intermediates, selecting these predictions might result in extra keystrokes. The cost of selecting prediction might go higher than the cost of actually typing out the word.

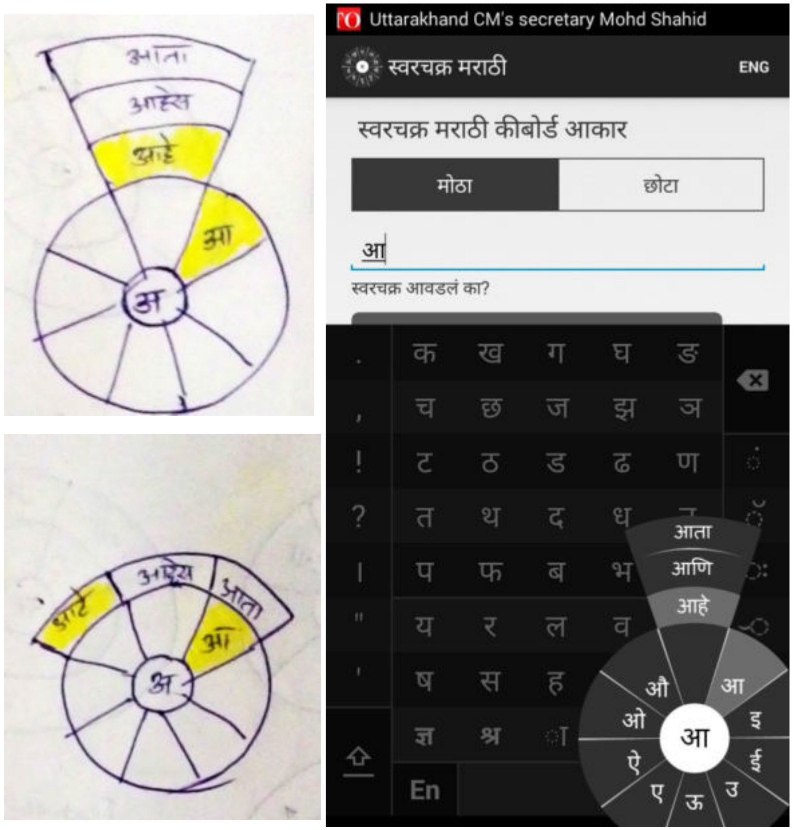


Fig. 17. Design Idea 1 | Bringing the standard prediction window from top of the keyboard to chakra

2.

Gesture typing for Swarachakra

It has been proven that shape typing boosts typing speeds for English. There are keyboards for Indian languages which also uses similar gesture typing. This design idea tries to extend and use the way user selects matras from the chakra. Here, the user will select a particular matra of a character to form a swarachakra unigram[1], once user selects a matra, next swarachakra unigrams considering the previous unigram will be predicted around the chakra as shown in image [Fig. 18]. Once user selects one of the predicted swarachakra unigram, next predictions will appear in the previous chakra. This will continue to form a word. At any point if a user doesn't find a relevant prediction, he can release the flow. Flow will again start after the user types the intended character. The flow of gestures of a user to form a word will be as shown in image [Fig. 18].

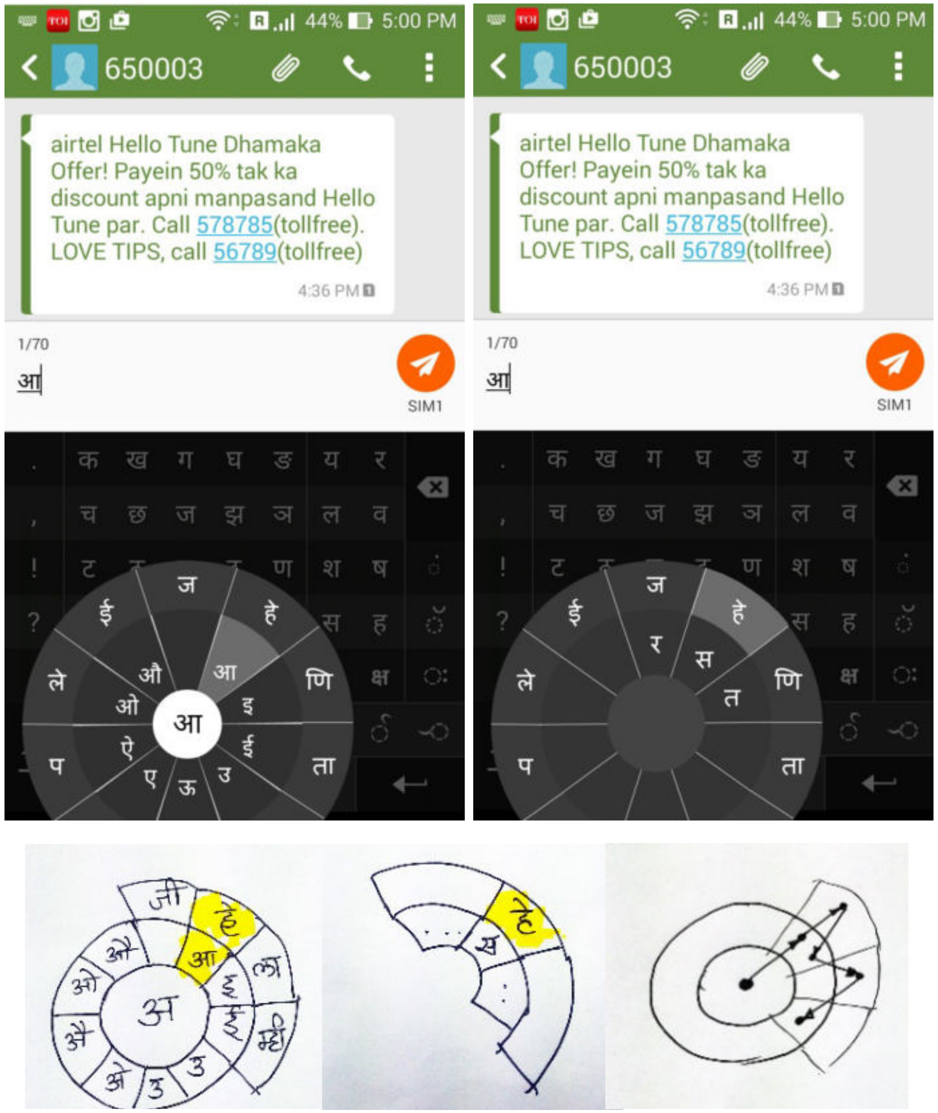


Fig. 18. Design Idea 2 | Gesture typing for Swarachakra

Swarachakra unigram is formed by a consonant and a vowel modifier like का is a swarachakra unigram which is formed from क and ा.

Limitations:

1. Selection of predicted characters:
Predicted characters are placed on the periphery of swarachakra. User needs to select a character from the prediction. The interaction to select the character is a pain point. User makes a gesture from inner circle to outer, to know that the user's intention is to select the character he is hovering on right now, a delay must be introduced. As there is a probability that the user is still making the shape and not on the final character and is just waiting on a character to read the other predicted characters but system might consider it to the intended selection and push the new chakra. This is an error case.
2. No Error correction: If a user commits an error while typing as mentioned in point 1, for user to come out of it, he/she has to erase the previous selection and type again. There is no way he/she can come out of the error in the flow. This decreases the typing performance of the user.
3. The delay introduced for making the selection, also hinders the typing performance.
4. The thought behind this idea was to build something analogous to flow or gesture in English. But the idea for english works, as the user swipes over keys, which are known and are in their muscle memory. Whereas, here the prediction position of characters is not known, a character appearing at one position might appear at some other based on the pre typed text.
5. For Beginners, This prediction might help as they anyway hunt for the next character on keyboard. This will reduce their hunting time. But for intermediates and experts, they will have have to look and scan the entire prediction list for every selection. In turn also hampers typing performance.
6. Thumb problem: User's thumb on the character will also make reading the predicted characters difficult. He cannot move his thumb, because if he moves his thumb, that might push a new predicted character chakra which a user don't want in the case.

3.

Next word highlight

This idea tries to solve the problem of Novice users. Novice users have problems discussed in section [Understanding Users]. They usually hunt and pick the alphabets from keyboard. They are so overwhelmed with keyboard itself, adding new concept of prediction and asking them use prediction will put a huge cognitive load on them. For these set of users, this predictive interface predicts the next probable alphabets provided they have selected one or many alphabets. In this way, the hunting activity on the keyboard for next key will reduce. It can compensate for a smaller prediction window and include highlights from shadow. As an when user selects the predicted character, based on the typed text predicted words are predicted in the traditional prediction bar.

Limitations:

The important limitation of this idea is when a user wants to type something, he types a character and the next intended character is not highlighted. Then there is strong visual disturbance of the predicted characters and finding the intended character becomes more difficult.

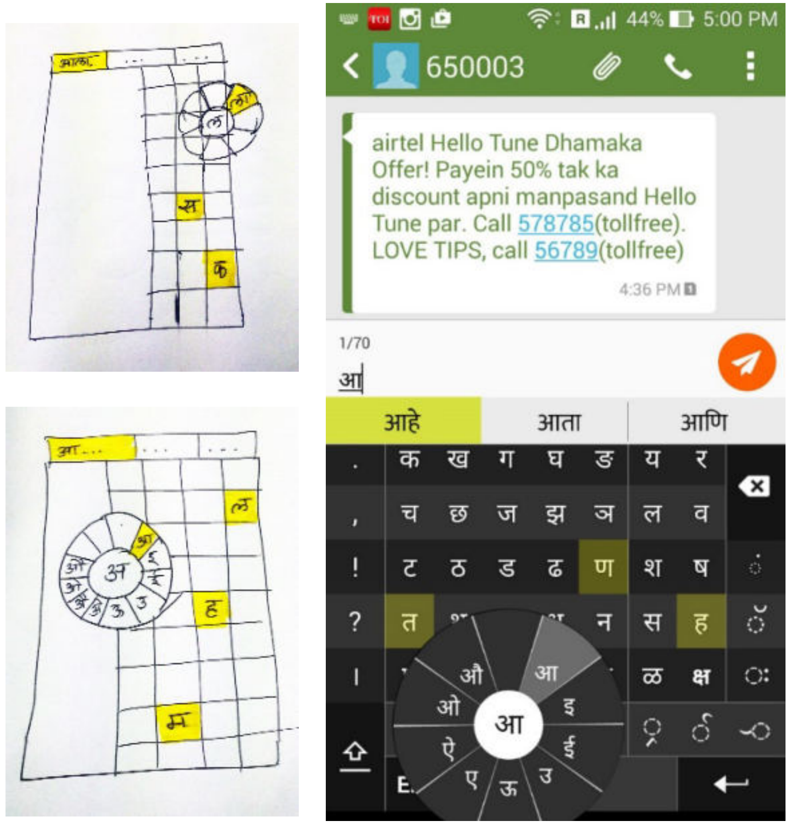


Fig. 19. Design Idea 3 | Next word highlight

4.

On Key Prediction

This is a concept in which the predicted words are shown on the key itself instead of the top bar. If a user selects a character, predictions provided that character will be predicted on respective keys. Here, If the prediction is not the there, then still user's eyes has to traverse to the next character he wants to type. Now, he can visually see if the prediction is there. If the word is predicted he selects it there itself, else he presses the next alphabet key. This prediction technique will help all kinds of user.

Limitations

1. It cannot handle the scenario when multiple high frequent words lie on the same key. As there is only one prediction per key
2. Prediction of long words on adjacent keys will create a problem.

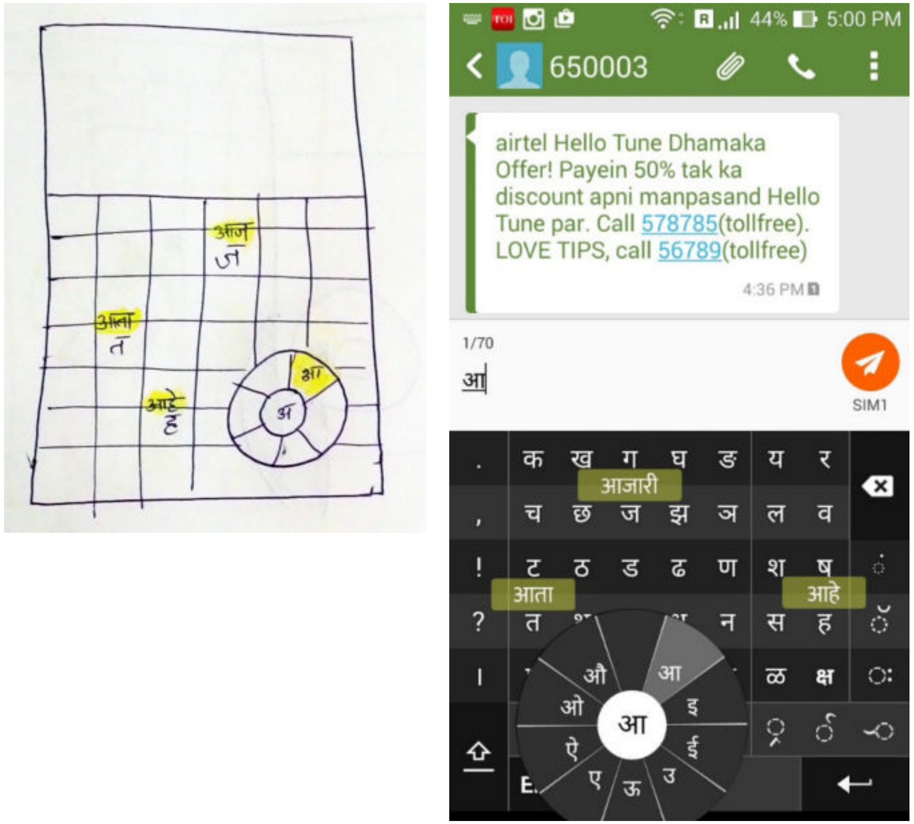


Fig. 20. Design Idea 4 | On Key Prediction

5.

Prediction using multiple single keystrokes

Users taps on the keys present in the desired word. For Novices, this prediction text entry method helps as it takes of the language rules which they are struggling with. Prediction algorithm will take care of all their errors. For intermediates and experts, it helps in increased typing speed.

अ + ह = आहे, आह
अ + ह + त = आहेत, आहात
ह + त = हात, हत्ती

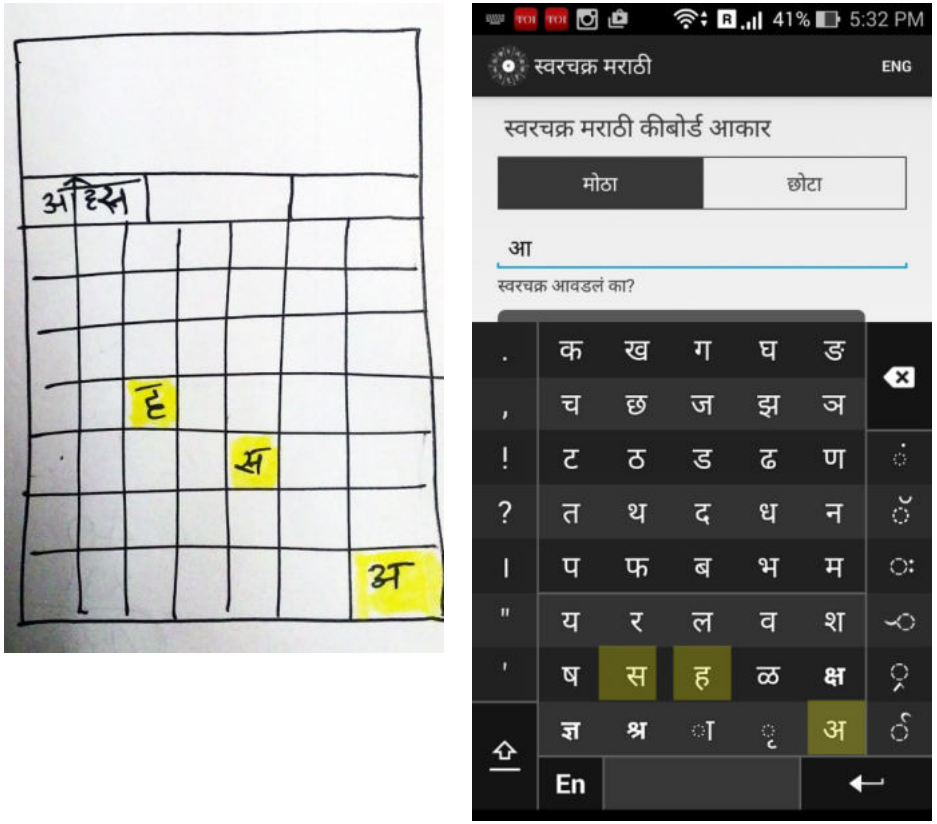


Fig. 21. Design Idea 5 | Prediction using multiple single keystrokes

11. Design Approaches

The approaches which emerge out of the observations seen above are:

Predict Less Approach

As observed above, conceptual model making of a predictive system is something which requires several hours of typing. Having a correct conceptual model of which words will be predicted by the predictive system is an expert behaviour. Even after you become an expert for a keyboard and have a conceptual model of predictions, there is still a small bit of uncertainty of the predictions. Thresholding the corpus, words having frequencies beyond this will be only predicted. This will try and remove the uncertainty parameter of pre diction system. User can create conceptual model of the system faster.

Problems involved in Predict less approach:

By this approach we try to build a muscle memory of most frequent words. So, a particular word will appear at a specific position. That means, the prediction will be static and will not depend on the context of message. Static prediction will have its own set of limitations. Static predictions cannot comply to regional variance of dialects of a language. As static predictions does not consider personal corpus of the user, the previous words typed in a statement, they are non-contextual and non-smart predictions.

Prediction which enables inflection (N-gram Approach):

As we saw, most Indian languages are agglutinative. Marathi is case for the project, it is an agglutinative language. Many inflections are possible from a part of the word. For example, in the picture below, a word घराच्याखाली have घरा + च्या + खाली. We can see in the example, with घरा as suffix so many infections are possible.

घराच्याखाली =	घरा	+ च्या +	खाली
	बिल्डिंग	+ च्या +	वर
	मंडपा		समोर
	झाडा		मागे
			पाठी
			आत
			बरोबर

In this approach of prediction, we use this property of language, and instead of predicting the entire word, we predict each n-gram. So, based on the character typed, only the n-gram would be predicted. Then, provided that n-gram, next n-gram would be predicted. By this way, we can reach out to more corpus and the predictability will go high.

Prediction Chakra Approach

This is a new interface design approach for prediction for Swarachakra. As, in swarachakra, we press a character and all the vowel modifiers of that character appear in the chakra. Prediction Chakra is an approach similar to this, in which we show the predictions in a chakra to maintain continuity. Also, prediction chakra should pushed on the keyboard itself, as it should account for the shift of attention problem. In this approach, user should be able to type and select predictions with seldom looking at the editor. We speculate, this approach will increase typing performance.

Predict complete words

Extra space keystrokes: It has been observed that large number of the corpus is covered by spaces. To reduce this keystroke of space, most of the predictive systems predict entire words.
Using the same prediction model of English: Most of the prediction mechanisms in Indian languages are borrowed from English. As for English, mostly frequency based complete word prediction is used. We use the same method.
Corpus is in that way: As the corpus we have for Indian languages does not have information about morphology and inflection. Severe language processing of corpus needs to be done to get a corpus which is language sensitive.

12. Design Ideas

Based on these approaches, various combinations of approaches can be a solution to the problem. The following section describes these combinations and identify which will work best for the problem defined within constraints.

The components used in the design ideas are explained below

Traditional prediction bar:
This is the prediction bar which is displayed on top of keyboard.

Fig. 27. Traditional prediction bar
Source[21]

N-gram prediction:
This is type of prediction in which we don’t predict complete words, instead predict n-grams of the word as discussed in section [Design Process II | Towards solution].
Complete Word Prediction:
This is an approach which is used traditionally, in which we predict the entire word based on the user input.

Prediction Chakra as prediction window:
This is an interface of prediction window in which predictions are displayed in a chakra similar to swarachakra.

Fig. 28 Prediction chakra

Entire Corpus:
Here by entire corpus we mean considering the entire corpus available for prediction

Less Corpus:
Here the corpus is thresholded to the pre-decided corpus and this corpus is used for prediction.

The three major considerations to iterate upon design ideas were, **When to predict? How to Predict? and How much to predict?**

Based on these considerations, the following questions are possible:

What to predict?	Where to predict?	How much to predict?
Complete word N-gram	Traditional on top of keyboard Prediction Chakra	Entire corpus Thresholded corpus

of each other. Answers to these questions can be combined in different ways to arrive at a solution. There is no intuitive way to know which combination may work.

Following are the various combinations based on the above questions:

1. N-gram prediction, Traditional way of displaying prediction and using entire corpus:

Here the number of keystrokes required to type in a word is increased than that of traditional complete word prediction. For example, in case of झाडावरदेखील, user has to select झाडा first, followed by वर and then देखील, where instead he could have selected झाडावरदेखील in one stroke. Hence, the number of keystrokes go high. But on the other hand this increases predictability. Probability of getting झाडावरदेखील when झा is typed is low than getting it if the typed text is झाडावर. Here for this approach, the speculation is using the traditional prediction window will make more sense, as the predictions will be appearing sequentially. So, user has to just tap and select the next predictions. If the word is not predicted completely, No space should be appended. Hence, extra keystroke of space user has to perform.

2. N-gram prediction, Traditional way of displaying prediction and using thresholded corpus:

The problem of shift of attention can be solved but with an assumption that the user has formed conceptual model of predictions as the corpus is small and hence he/she won't look for predictions which he/she is sure won't be predicted. So, they don't have to dodge between keyboard and prediction window. As the number of words in the corpus are less, there is a chance that user builds a muscle memory of the position where the prediction will appear. This might boost the typing performance of user.

3. N-gram prediction, Prediction chakra for displaying prediction and using entire corpus:

In in this approach, the predictions are appearing in the prediction chakra as shown in [fig. 28] To call the prediction chakra, user has to drag from a character outwards in swarachakra. Once, he crosses a threshold the prediction chakra will be called and the predictions of that character will be shown. For n-gram prediction, the user has to drag and see the predicted n-grams, select one and again drag and see the next predicted n-grams (Drag and see, drag and see, drag and see). Speculation is this is not a right paradigm for n-gram predictions, as the user after getting one prediction has to just select the next predictions. So, the n-gram prediction does not make much sense here is a speculation.

4. N-gram prediction, Prediction chakra for displaying prediction and using thresholded corpus:

Again here, as mentioned in above the approach would be Drag and see, drag and see, drag and see. The idea of predicting less corpus was to build a conceptual model of predictions in user's mind and getting the predictions in the muscle memory of users. Remembering the prediction position and getting it into muscle memory is a bit difficult as the n-gram prediction would be highly dynamic for Marathi (being an agglutinative language). The position of an n-gram might change depending on the previous n-gram.

5. Word prediction, Traditional way of displaying prediction and using entire corpus:

This is the default model which is used by all current predictive systems. The problems with this model are already discussed in section [Understanding problem]

6. Word prediction, Traditional way of displaying prediction and using thresholded corpus:

In this approach, as we are using thresholded corpus, the problem of shift of origin might be solved and having a muscle memory of frequent words also might be possible as discussed in 2nd design idea mentioned above. The predictions would be static.

7. Word prediction, Prediction Chakra for displaying prediction and using entire corpus.

The problem of shift of attention due to visual discontinuity might be solved as the prediction is happening right on the keyboard. Whereas, entire corpus still brings that uncertainty component up and building the conceptual model of predictive system will be a problem. Entire word prediction makes sense in prediction chakra paradigm as the user will be making a gesture to select a word. Whereas, as the entire corpus is included getting the prediction into muscle memory would be a problem.

8. Word prediction, Prediction Chakra for displaying prediction and using thresholded corpus.

This approach will have the advantages of word prediction and prediction chakra as mentioned above in point 7, but also will help to build fast conceptual model and users can have predictions stored in their muscle memory. So, users will store gestures of frequent words.

13. Final selected concepts

Comparing the combination of approaches mentioned above. The final idea which within constraints which will be designed and developed was to use a Prediction chakra approach with thresholded corpus and complete word prediction. We address the the shift of attention problem by predicting right there on keyboard where user is typing.

By predicting less , we thresholdize corpus to most frequent top 1000 words which covers almost 50% of the corpus. One can always argue on the fact that, prediction system should predict what user wants and he/she might be wanting a word from other 50% of the corpus. But is selecting the word from the rest 50% of the corpus account for the cognitive toll. There are multiple ideas behind this design decision one is to help users build this conceptual model fast. Less words in the corpus, hence less uncertainty. The other is to make the predictions go in the muscle memory of users. As the idea is to keep predicting specific word at specific position. User will over the time have a muscle memory of that word. This will reduce the hunting of the predicted word. In turn will improve the performance.

As discussed in above section, the speculation is for prediction chakra paradigm, entire word prediction is a better approach. Of-course, we cannot say it with any confidence without evaluating the idea.

Interface Details:

When a user taps on a consonant, he drags to the vowel modifier to attach a modifier. If a user further drags his finger and crosses a predefined threshold, predictions of that swarachakra unigram = consonant + vowel modifier will be displayed in the prediction chakra. There will be 3 predictions in the prediction chakra as shown in figure below.

If a user goes to the prediction chakra and discovers the prediction is not relevant, he can simply release his finger or drag further. By this, the original Swarachakra unigram will be selected. No space will be appended if user does not select prediction. If a user selects prediction, an automatic space will be appended to the selected text from prediction.

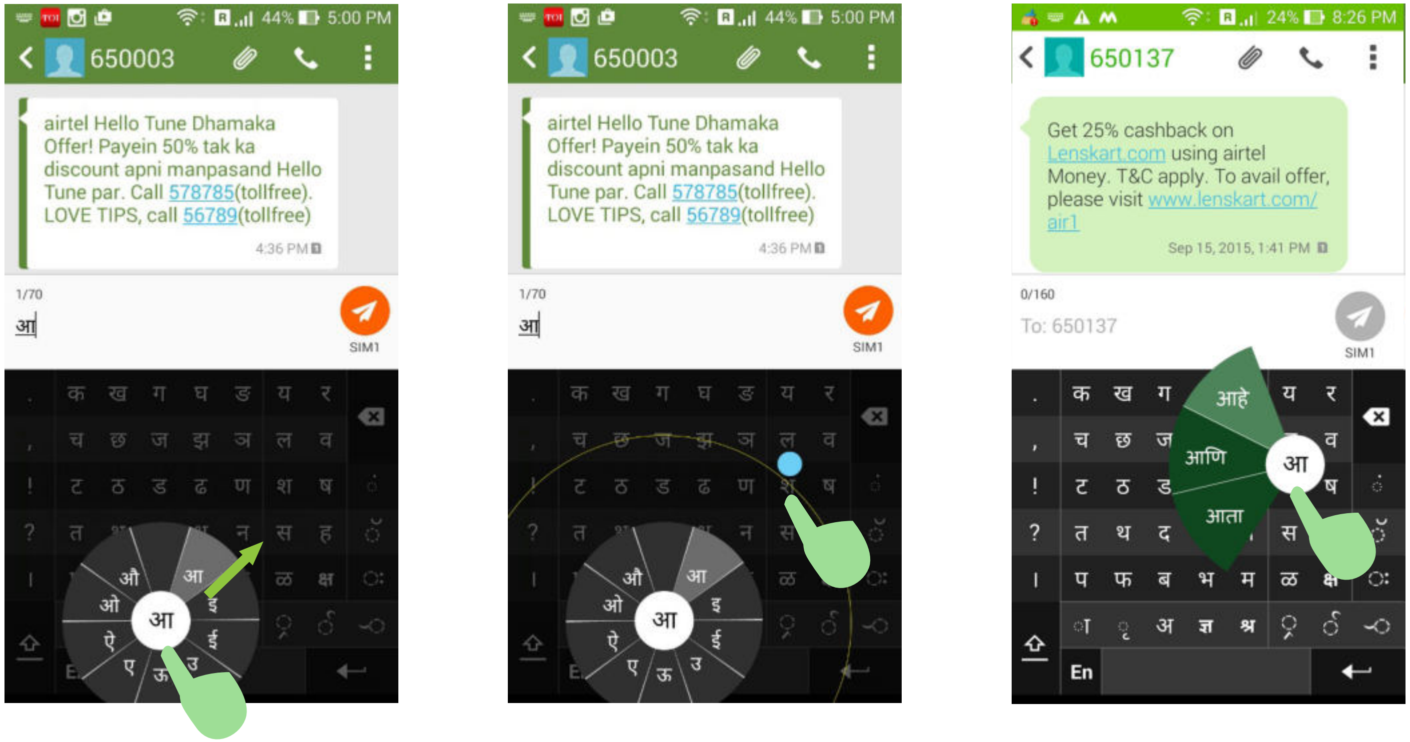


Fig. 29. Design idea interaction

If there was no relevant prediction in the first place, user can also try the prediction on further characters of word. The predicted list will be sorted and predictions of words provided the previously typed text will be shown in prediction chakra.
For example, if निवारा is the word a user wants to type, after typing नि if the word is not there in the prediction window. User can type the next character वा and trigger the prediction chakra of निवा.

If a swarachakra bigram which a user selects lies in the left half of the chakra, the predictions will appear in right half and vice versa as shown below. This is to account the visual obstruction caused by the finger when you select a matra.

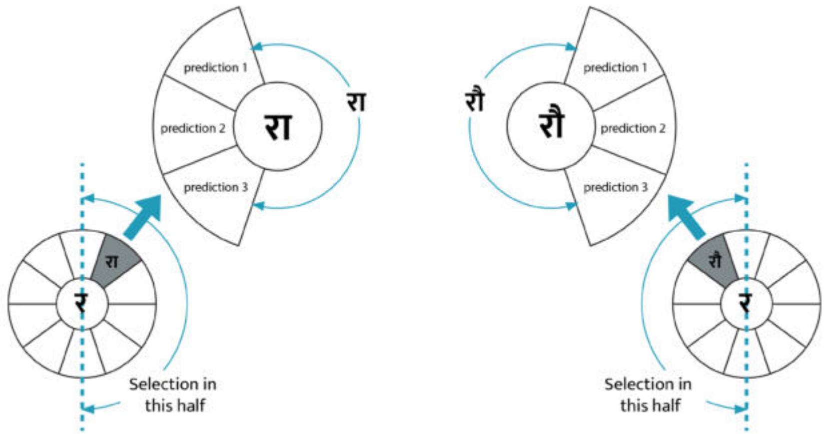
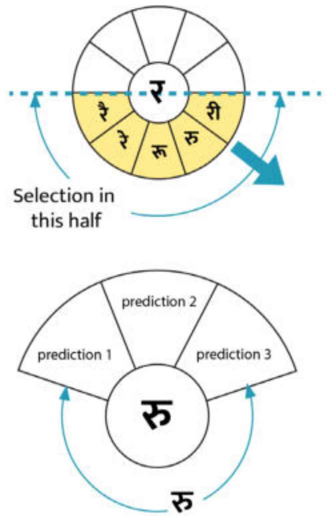


Fig. 30. Prediction chakra of keys based on the direction of selection

As the corpus is thresholded, there will be instances where there is no prediction in the prediction chakra.

For the keys which are at the bottom of the keyboard as shown in image below[Fig. 31]. If the selection is in the lower half of the swarachakra, then the predicted words won't appear the way shown before, but the predicted words in the prediction chakra would be like image[Fig. 31].

Special case of bottom keys



Prediction chakra of bottom keys

Fig. 31 . Prediction chakra of special keys

The images below are few screenshots of the interface developed for this project. The red mark shows the position when the prediction chakra be seen. From images you can see, if the user drags his finger from a matra, beyond a pre-defined threshold prediction chakra is called. The immediate adjacent screenshot shows the prediction chakra for the selection. As, you can see, there are few instances where there is no prediction in the chakra. This is because of thresholded corpus.

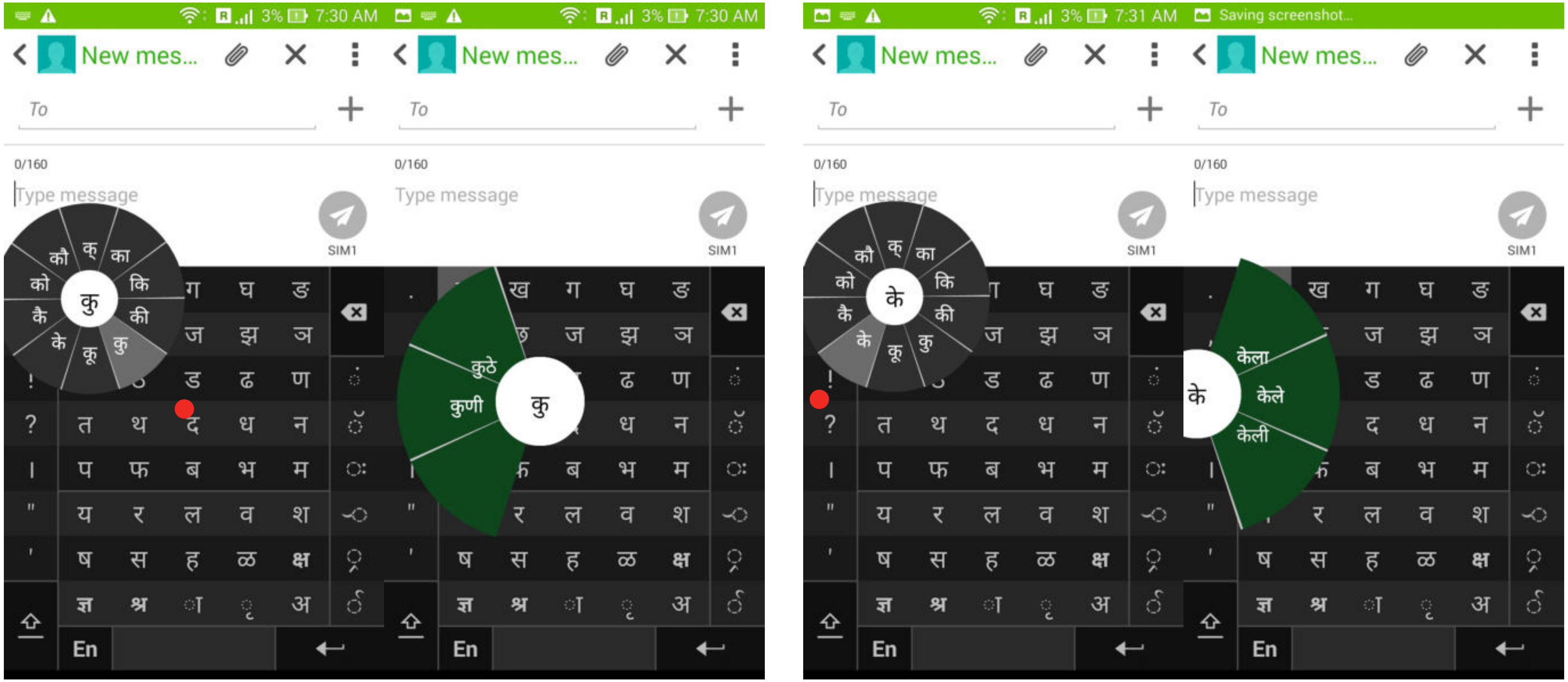


Fig. 34 (a) . Screen-shots of prediction interface

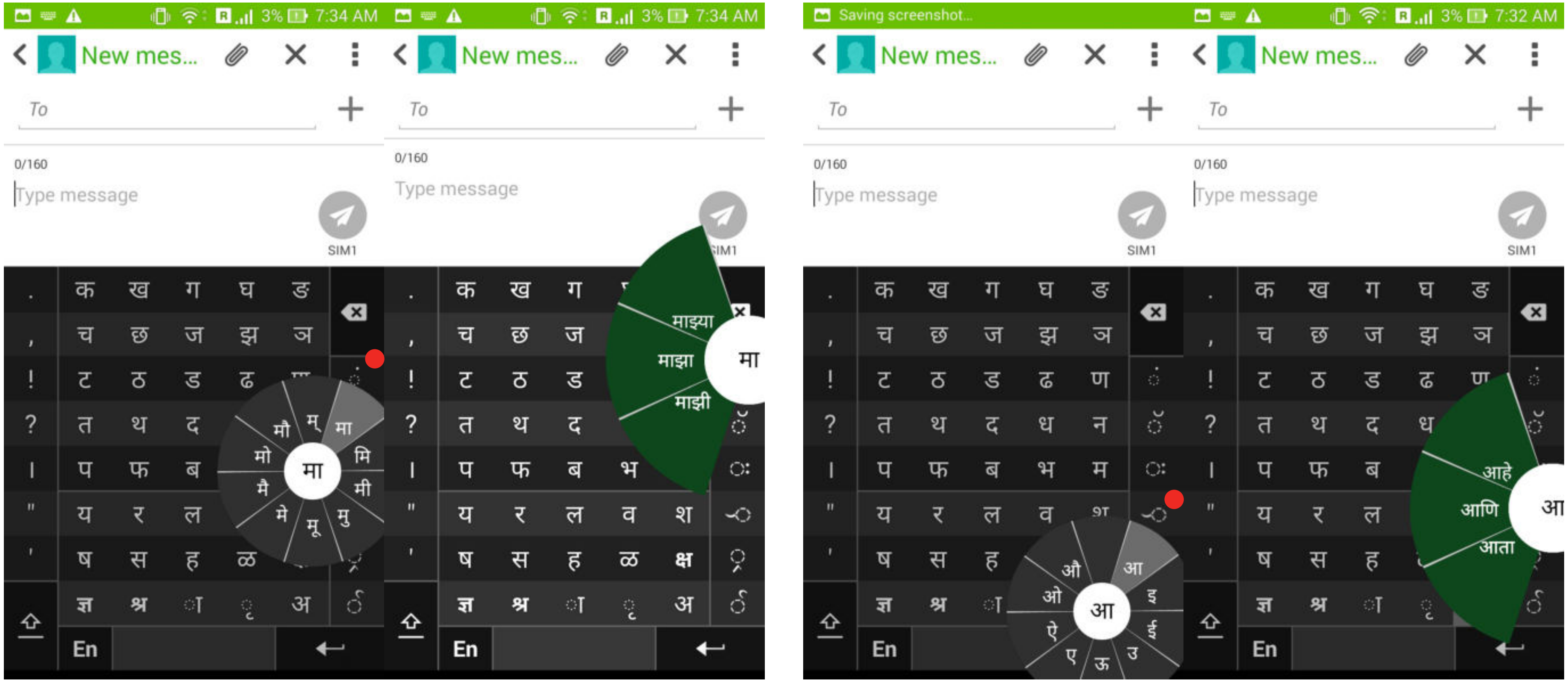


Fig. 34 (b) . Screen-shots of prediction interface

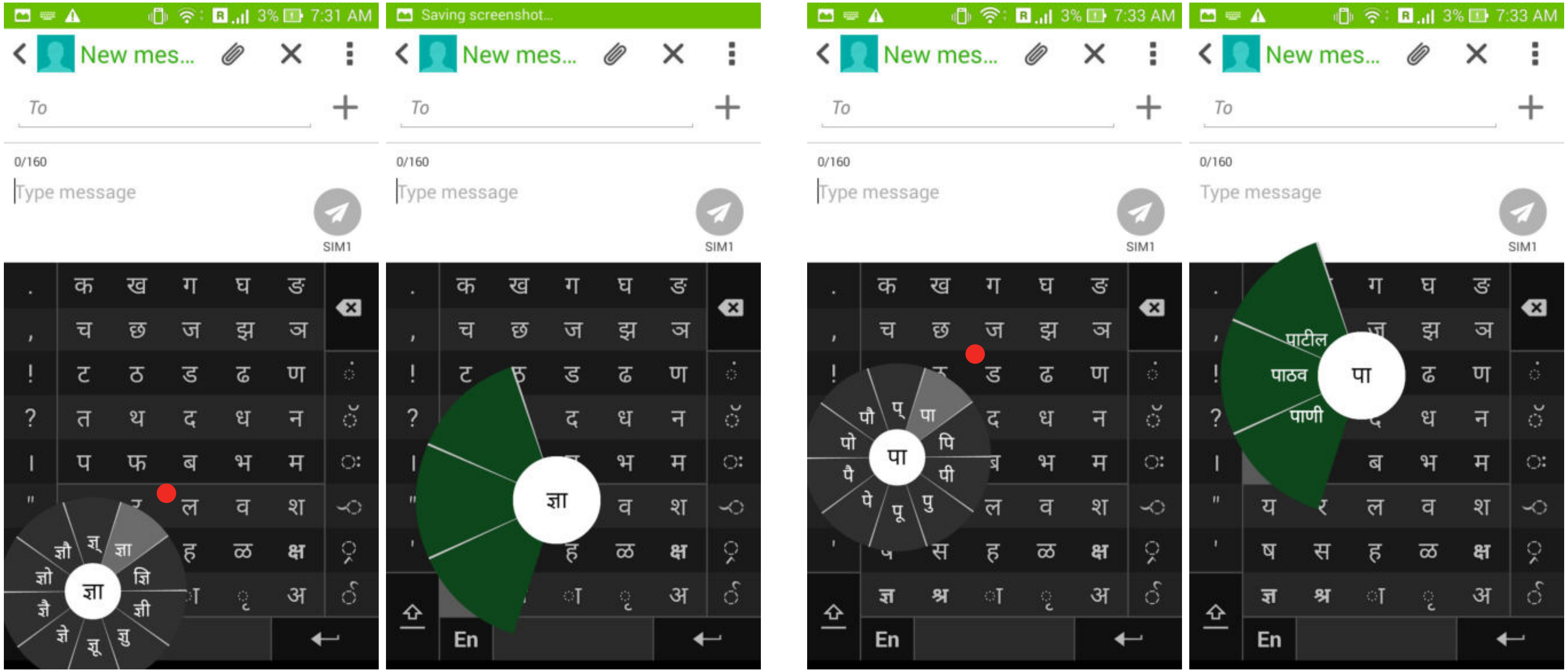


Fig. 34 (c) . Screen-shots of prediction interface

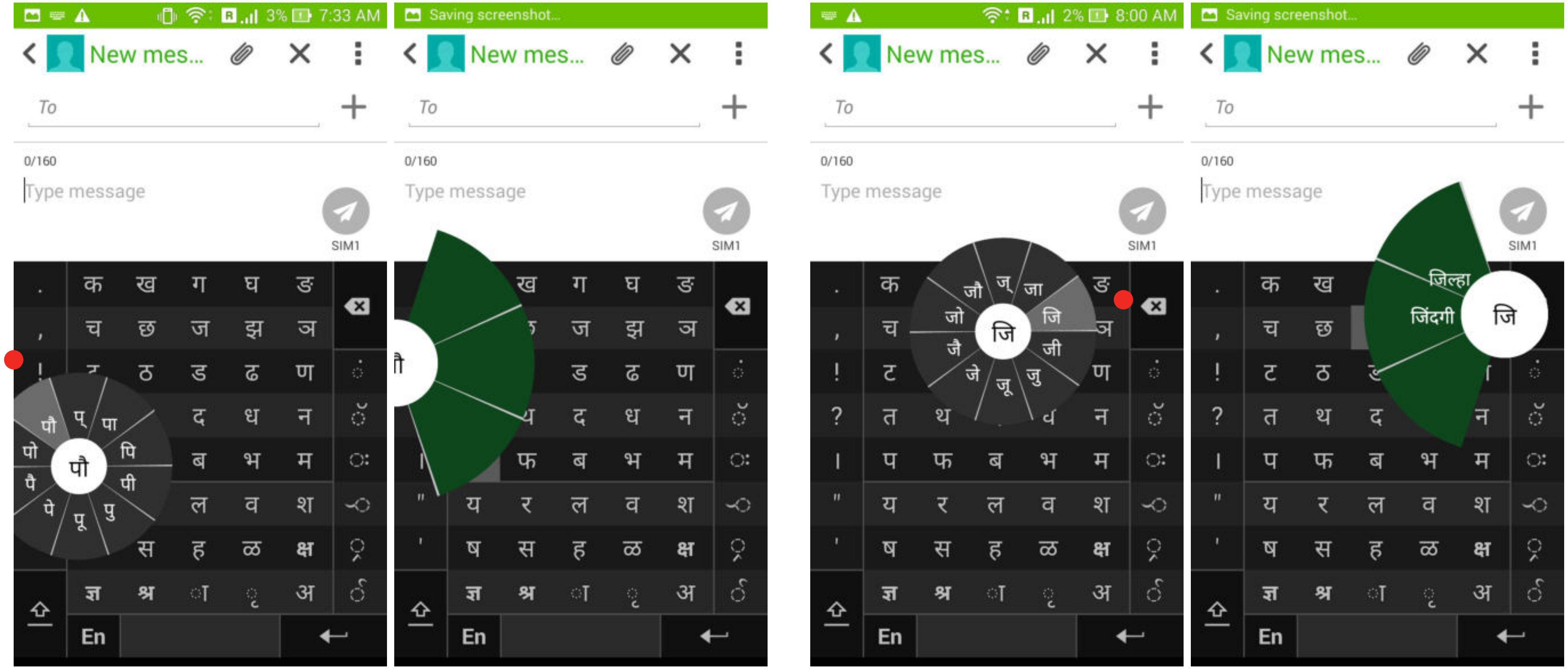


Fig. 34 (d) . Screen-shots of prediction interface

14. Limitations of Design

1. Words with no matras (vowel modifiers) won't be predicted .
2. For extremities, calling the prediction chakra is a challenge as shown in diagram below.

The chakra pop-ups at various extremes are shown below. The extremes of the keyboard are-

Fig. 32 . Keys with error cases

The regions for which the prediction chakra is difficult to call are shown in the yellow region in image [Fig. 33]

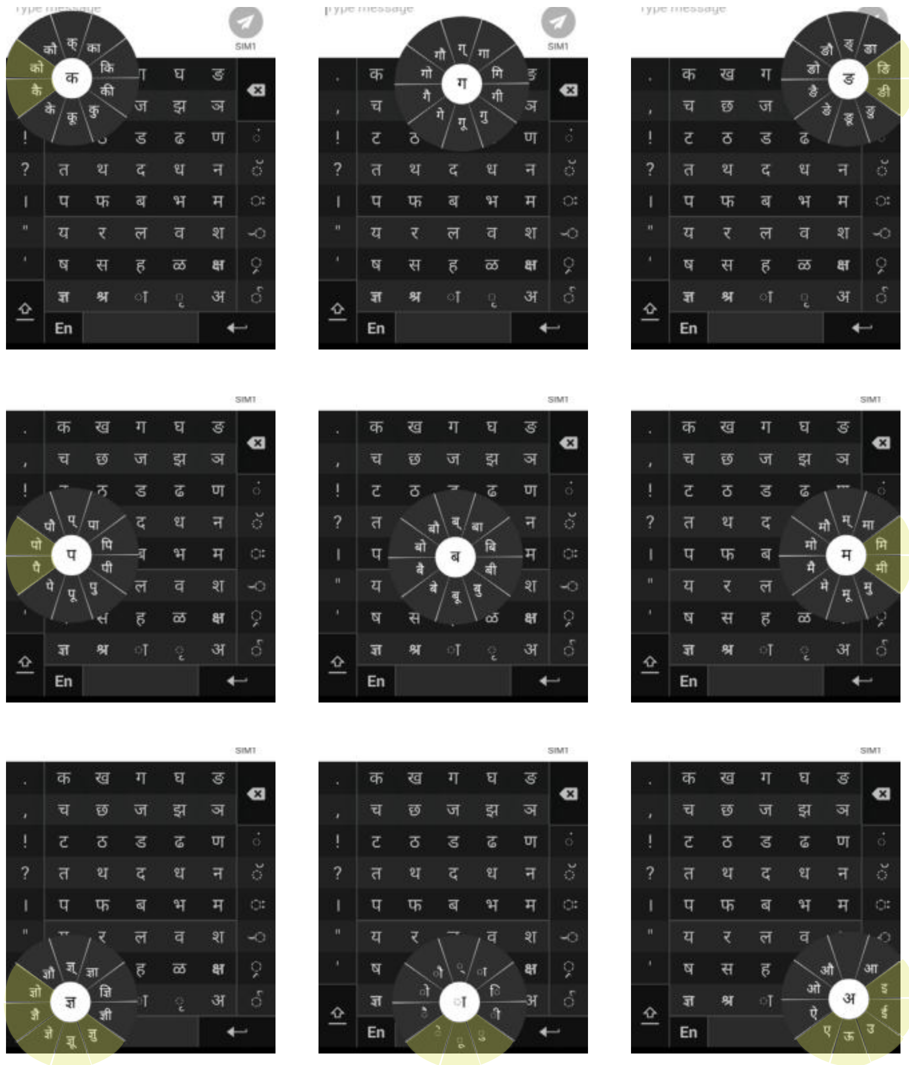


Fig. 33 . Error cases

Here are the all the characters, prediction of which cannot be called with current interface.

कै	को	ङि	ङी
चै	चो	ञि	ञी
टै	टो	णि	णी
तै	तो	नि	नी
पै	पो	मि	मी
यै	यो	शि	शी
षै	षो	क्षि	क्षी
ज्ञै	ज्ञो	इ	ई

ज्ञे	ज्ञू	ज्ञु
श्रे	श्रू	श्रु
ए	ऊ	उ

Table 1. List of alphabets predictions of which is not possible

The corpus coverage of these letters in words is found and described next. If the letter is present at the end of word, it is excluded as the prediction is not expected at that point. As, user must have already typed the entire word till then. Frequency count and the relative frequency is found for all these cases and described in the table below.

	Frequency	Relative frequency		Frequency	Relative frequency
कै	15,911	0.0004435	को	2,51,452	0.0070098
चै	6,339	0.0001767	चो	17,840	0.0004973
टै	589	0.0000164	टो	23,770	0.0006626
तै	3,244	0.0000904	तो	1,54,060	0.0042948
पै	40,791	0.0011371	पो	1,04,867	0.0029234
यै	1,590	0.0000443	यो	76,300	0.0021270
षै	230	0.0000064	षो	1,776	0.0000495
ज्ञै	97	0.0000027	ज्ञो	280	0.0000078

Table 2. Alphabets with their respective frequency and relative frequency

	Frequency	Relative frequency		Frequency	Relative frequency
ङि	893	0.0000248	ङी	1,578	0.0000439
ञि	4,772	0.0001330	ञी	5,476	0.0001526
णि	24,862	0.0006930	णी	53,553	0.0014929
नि	2,94,387	0.0082067	नी	68,589	0.0019120
मि	2,67,480	0.0074566	मी	79,125	0.0022058
शि	2,43,013	0.0067745	शी	67,429	0.0018797
क्षि	12,332	0.0003437	क्षी	4,181	0.0001165
इ	1,60,566	0.0044761	ई	2,49,950	0.0069679

Table 3. Alphabets with their respective frequency and relative frequency

	Frequency	Relative Frequency		Frequency	Relative Frequency		Frequency	Relative Frequency
ज्ञे	614	0.0000171	ज्ञू	83	0.0000023	ज्ञु	204	0.0000056
श्रे	5,630	0.0001569	श्रू	835	0.0000232	श्रु	2,222	0.0000619
ए	3,27,834	0.0091391	ऊ	1,77,037	0.0049353	उ	3,43,782	0.0095837

Table 4. Alphabets with their respective frequency and relative frequency

The total frequency count of these letters from the above table is 30,95,563 out of the total corpus which contains 3,58,71,284. This contributes to 8.629% of the total corpus. That means, the current prediction mechanism cannot predict these 8.629% of the words.

This 8.629% is not a true number. It is not the case that these 8.629% words will never be predicted by the system.For example, consider चोराच्या, as it starts चो, it will be listed in these 8.629% words of not prediction. But after चो user will type रा and this word can be predicted there as well, in the predition chakra of रा. So, there will be lots of words out of these 8.629% which the system can still predict even after having चो or any other error position in it.

As these above mentioned key positions are one of the limitation of this design. The other very important one is words without having matras. Words starting or having consonant in it cannot be predicted by this design.

For example, कमळ can never be predicted by this design as this word has all the consonants. But, कमाल can be predicted, not at the first place because consonant क cannot trigger prediction but मा can trigger as it has consonant. So, calculations should be done to find out what percentage of the words having just consonants are not getting predicted by this design.

Another important thing to consider before calculations is words starting with just a consonant, only the words which will be in prediction window should be considered, as the words in the shadow are anyways not predicted. This applies to all the consonants present at second, third upto (n-1)th place (n is length of the word).

15. Evaluation

A within subject evaluation was conducted with 5 users. Experiment was designed to test 3 keyboards, Swarachakra with predictions in prediction chakra and with less predictions (corpus coverage of 47.33%), Swarachakra with predictions in prediction chakra and with more predictions (corpus coverage of 79.88%) and Swarachakra without prediction. Out of 5 users, 2 were expert users of Swarachakra without prediction, 1 was intermediate user of Swarachakra and 2 were novices.

The primary objective of the experiment was to test which keyboard among the three performs best. User typing accurately and faster implies better performance. Will prediction will even work for Swarachakra?

More specific goals to test of the experiment were as follows:

- Will Prediction improve typing speed on Swarachakra?
- Will predicting less words will help in faster building of conceptual model of predictive system?

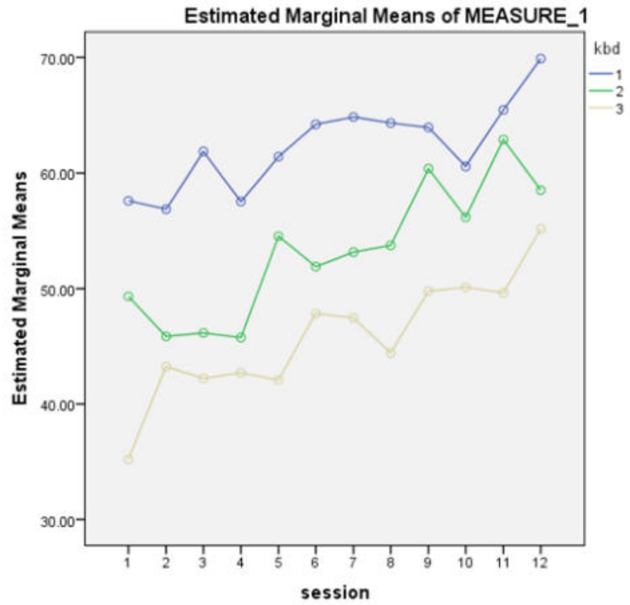
Evaluation Protocol

Each user had 3 attempts of each keyboard. In each attempt, s/he underwent 4 sessions of typing. In each session, user typed 20 phrases. The phrase selection was done in such a way that words with different typing complexity were included. The phrases are included in the appendix. A tool designed by Swarachakra team tool was used to capture various data like CPM, Accuracy, Edit distance, keystroke logs and time stamp.

16. Results

Every user attempted each keyboard thrice. In every attempt, user underwent 4 typing sessions. In all, every user underwent 12 typing sessions for each keyboard. Average CPM of each session was calculated. A two-way Anova was performed on the data gathered from the experiment. Below Figure 16 shows the Estimated Marginal Means of the CPM versus sessions for all users. The graph shows the Estimated Marginal Means for all the three keyboards.

It can be seen from the Evaluation that Swarachakra without prediction performed best among the three keyboards. Swarachakra with Less prediction performed better than Swarachakra with More prediction. The typing performance of users on all three keyboards is increasing as they are typing more. Few users from user testing were Expert users of Swarachakra. In the expert behavior it was observed that users tend to not look at the predictions at all. Probably this is one of the reason for the poor performance of predictive keyboards over non-predictive in evaluations because of which the evaluation data must have been skewed.



1. Swarachakra without Prediction
2. Swarachakra with Less Prediction (47.33%)
3. Swarachakra with More Prediction (79.88%)

Fig. 34 . Performance of keyboards measured in CPM across sessions

17. Conclusion

It was observed that Swarachakra without prediction performed best over other predictive Swarachakra keyboards. However, we speculate few reasons for the same. Few users in the evaluation were expert users of Swarachakra, which resulted in skewing of data. Also, the prototype used needed more refinement in terms of development, this might also be a potential reason for poor performance.

The future path of the project would be, screening of novice users for evaluation, as they all will be equally new to Swarachakra and prediction. This screening of users will reduce the data getting skewed in the evaluation. Because of the academic limitation of time for the project, the solution was evaluated for 3 days. The evaluation should be conducted for a longer duration as conceptual model building of the predictive system even of the thresholded corpus will take time. Even to test that predictions of most used words in muscle memory will take considerate time. The hypothesis can be tested only by a longitudinal evaluation.

18. References

1. Does Prediction Really Help in Marathi Text Input? Empirical analysis of a Longitudinal Study , MobileHCI 2016 (Forthcoming)
2. Denis Anson MS and OTR , Penni Moist OTR , Mary Przywara OTR , Heather Wells OTR , Heather Saylor OTR & Hantz Maxime OTR The Effects of Word Completion and Word Prediction on Typing Rates Using On-Screen Keyboards[Journal]
3. Sheri Hunnicutt*, Lela Nozadze†, George Chikoidze Russian Word Prediction with Morphological Support
4. Anirudha Joshi, Girish Dalvi, Manjiri Joshi: Corpus of Marathi Word Frequencies from Touch-Screen Devices Using Swarachakra Android Keyboard
5. Anirudha Joshi, Girish Dalvi, Manjiri Joshi, Prasad Rashinkar, Aniket Sarangdhar: Design and Evaluation of Devanagari Virtual Keyboards for Touch Screen Mobile Phones
6. Time.com techland.time.com/2011/07/21/study-fewerthan-50-of-smartphone-users-make-calls/
7. https://lh3.googleusercontent.com/Rc94sUxmtpZ1Ii-6OOsI6xxlrZw6uJb1kyUE3TsCS2kusPSq4k9Lo0CuC98HUAc4Devz=h900
8. http://i-cdn.phonearena.com/images/articles/127909-image/How-to-turn-off-SwiftKey-keyboards-overzealous-text-prediction-and-corrections.jpg
9. http://www.mforum.ru/cmsbin/2011/26/wp7-sliding-keyboard-swype_full556x450.jpg
10. http://4.bp.blogspot.com/-Ygs4o-_ApkI/Ufrz6S2O6zI/AAAAAAAACqQ/qQDozxA_jo0/s1600/2013-08-02+08.23.17.png
11. http://www.gizmodo.jp/upload_files2/20120514fake_keyboard_of_bb10_for_iphone.jpg
12. http://www.geek.com/wp-content/uploads/2012/05/bb10-predictive-keyboard.jpg
13. http://www.berryreview.com/wp-content/uploads/2012/10/0_0_0_http-__i.haymarket.net_.au_News_BlackBerry10-_keyboard_thumb.png?11fac8
14. https://lh3.googleusercontent.com/gNiECEWhKe5UdS3-mk7yFKweEHsBQEjCMqxSlUR6JbtftsUuo4YlQW7nb1DtWhk-tLpU=h900
15. https://androidmarket.googleusercontent.com/android/market/com.eightpen.android.eightpen/ss-480-0-0
16. http://www.androidpolice.com/wp-content/uploads/2012/12/nexusae0_110.png
17. https://lh3.ggpht.com/iydM9ZbATE6KaySkG0w6_bPeCxLv3Ome0FiNx5ZXgOyXuYxu-I4ToDrTCcXbRItfTMON=h900
18. https://ploum.net/images/messagease.png
19. http://ndl.mgccw.com/mu3/app/20140329/16/1396081656500/sss/0_small.png
20. https://blog.swiftkey.com/content/uploads/2015/05/MarathiLayouts-SwiftKeyAndroid.png
21. http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2013%20Reports/Mobile-Consumer-Report-2013.pdf
22. http://yourstory.com/2015/07/mobile-internet-report-2015/
23. http://gadgets.ndtv.com/mobiles/news/india-overtakes-china-in-high-end-smartphone-volume-growth-hsbc-713831
24. https://help.gnome.org/users/dasher/unstable/basics.html.en
25. http://www.phonearena.com/news/MessagEase-keyboard-improves-your-typing-speed_id43609
26. D., & Joshi, A. (2013). Technology adoption by 'emergent' users. Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction - APCHI '13. doi:10.1145/2525194.2525209
27. Quinn, P., & Zhai, S. (2016). A Cost-Benefit Study of Text Entry Suggestion Interaction. Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16. doi:10.1145/2858036.2858305

19. Appendix

Phrases for Keyboard having Lesser Predictions

चिमणी करते चिव चिव
किती वेळ लागेल
जोवरी पैसा तोवरी बैसा
तू कसा आहेस
राव चढले पंत पडले
काळाकाळा कापूस पिंजला रे
भारतभाग्यविधाता
मामाची बायको सुगरण रोजरोज पोळी-शिकरण
नदीनाल्यांना आला पूर
कामापुरता मामा आणि ताकापुरती आजी
झोंबे अंगा वारे काया थरथरे
पळस गेलं कोकणात तीन पानं चुकेनात
खायला कोंडा नि निजेला धोंडा
पावसाच्या रेघांत खेळ खेळू दोघांत
सारे भारतीय माझे बांधव आहेत
पुढे मला काही कल्पना सुचू लागल्या
परहित आधी नंतर स्वहित साधावे
कराग्रे वसते लक्ष्मी करमध्ये सरस्वती
दैव देते आणि कर्म नेते
पचापचा शिव्या देई खाताखाता पान

Phrases for Keyboard having No Predictions

झाली सकाळ सरली रात
ससा ससा दिसतो कसा
बरेच ढग दिसत आहेत
मला थोडे पाणी देता का
रमेश जेवण कर
बागेभोवती भिंत आहे
आठवण आहे ना तुला
घरी सगळे कसे आहेत
पावसाच्या रेघांत खेळ खेळू दोघांत
विंध्य हिमाचल यमुना गंगा
ते माझं कौतुक करू लागले
काखेत कळसा गावाला वळसा
एका छान अनुभवाला तो मुकला होता
नाच रे मोरा आंब्याच्या वनात
धुरांच्या रेघा हवेत काढी
डेबू मामाच्या शेतावर कष्ट करू लागला
पाखरे घरट्यांत जाऊन बसली आहेत
कुलस्त्री जसे हास्य ओठात शोभे
स्वतः मेल्याशिवाय स्वर्ग दिसत नाही
स्वराज्य हा माझा जन्मसिद्ध हक्क आहे

Phrases for Keyboard having More Predictions

पण लक्षात कोण घेतो
हवेत उडतो लाल लाल फुगा
तू कशी आहेस
घरी कशी मग सांगा जातिल
माकडाने रंगवले आपले तोंड
आपण कुठून आलात
टपटप पानांत वाजती रे
हात लावता पंख फाटतिल
झोंबे अंगा वारे काया थरथरे
मीना गोष्ट वाचत होती
दोघांनीही आपापले पैसे मोजले
आभाळात छानछान सातरंगी कमान
कुठूनही गेले तरी पोरांची नजर पडणारच
तुझंमाझं जमेना तुझ्यावाचून करमेना
सरड्याची धाव कुंपणापर्यंत
पाखरे घरट्यांत जाऊन बसली आहेत
सुधेसारखा साद स्वर्गीय गाणे
स्वतः मेल्याशिवाय स्वर्ग दिसत नाही
चार आण्याची कोंबडी अन बारा आण्याचा मसाला
दुपारी चारच्या सुमारास पाऊस सुरू झाला

These were the phrases used during evaluation for all the three keyboards. Phrases with different complexities were selected for the evaluation.

	Words	Word Count	No. of words Predicted	Words (1 = word predicted and 0 = word not predicted)					
	पण लक्षात कोण घेतो			पण	लक्षात	कोण	घेतो		
Prediction after					लक्षा				
		4	1	0	1	0	0		
	हवेत उडतो लाल लाल फुगा			हवेत	उडतो	लाल	लाल	फुगा	
Prediction after				हवे					
		5	1	1	0	0	0	0	
	तू कशी आहेस			तू	कशी	आहेस			
Prediction after				तू					
		3	1	1	0	0			
	घरी कशी मग सांगा जातिल			घरी	कशी	मग	सांगा	जातिल	
Prediction after				घरी					
		5	1	1	0	0	0	0	
	माकडाने रंगवले आपले तोंड			माकडाने	रंगवले	आपले	तोंड		
Prediction after						आपले			
		4	1	0	0	1	0		

Table 5 . Truth table of phrases used for evaluation of keyboard with More predictions

	आपण कुठून आलात			आपण	कुठून	आलात			
Prediction after						आला			
		3	1	0	0	1			
	टपटप पानांत वाजती रे			टपटप	पानांत	वाजती	रे		
Prediction after							रे		
		3	1	0	0	0	1		
	हात लावता पंख फाटतिल			हात	लावता	पंख	फाटतिल		
Prediction after				हा	लावता				
		4	2	1	1	0	0		
	झोंबे अंगा वारे काया थरथरे			झोंबे	अंगा	वारे	थरथरे		
Prediction after						वारे			
		4	1	0	0	1	0		
	मीना गोष्ट वाचत होती			मीना	गोष्ट	वाचत	होती		
Prediction after				मीना	गो		होती		
		4	3	1	1	0	1		

Table 5 . Truth table of phrases used for evaluation of keyboard with More predictions

Prediction after									
	दोघांनीही आपापले पैसे मोजले			दोघांनीही	आपापले	पैसे	मोजले		
Prediction after		5		दोघांनीही	आपा	पैसे			
			3	1	1	1	0		
	आभाळात छानछान सातरंगी कमान			आभाळात	छानछान	सातरंगी	कमान		
Prediction after		5			छानछा		कमा		
			2	0	1	0	1		
	कुठूनही गेले तरी पोरांची नजर पडणारच			कुठूनही	गेले	तरी	पोरांची	नजर	पडणारच
Prediction after		5			गे	तरी			
			2	0	1	1	0	0	0
	तुझंमाझं जमेना तुझ्यावाचून करमेना			तुझंमाझं	जमेना	तुझ्यावाचून	करमेना		
Prediction after		7							
			0	0	0	0	0		
	सरड्याची धाव कुंपणापर्यंत			सरड्याची	धाव	कुंपणापर्यंत			
Prediction after		6							
		3	0	0	0	0			

Table 5 . Truth table of phrases used for evaluation of keyboard with More predictions

	पाखरे घरट्यांत जाऊन बसली आहेत			पाखरे	घरट्यांत	जाऊन	बसली	आहेत	
Prediction after				पाखरे			बसली	आहे	
		5	3	1	0	0	1	1	
	सुधेसारखा साद स्वर्गीय गाणे			सुधेसारखा	साद	स्वर्गीय	गाणे		
Prediction after						स्वर्	गाणे		
		5	2	0	0	1	1		
	स्वतः मेल्याशिवाय स्वर्ग दिसत नाही			स्वतः	मेल्याशिवाय	स्वर्ग	दिसत	नाही	
Prediction after						स्वर्		ना	
		5	2	0	0	1	0	1	
	चार आण्याची कोंबडी अन बारा आण्याचा मसाला			चार	आण्याची	कोंबडी	अन	बारा	आण्याचा
Prediction after				चा		कोंबडी		बारा	
		7	4	1	0	1	0	1	0
	दुपारी चारच्या सुमारास पाऊस सुरू झाला			दुपारी	चारच्या	सुमारास	पाऊस	सुरू	झाला
Prediction after				दु		सुमा		सुरू	झाला
		6	4	1	0	1	0	1	1

Table 5 . Truth table of phrases used for evaluation of keyboard with More predictions

	Words	Word Count	No. of words Predicted	Words (1 = word predicted and 0 = word not predicted)					
	चिमणी करते चिव चिव			चिमणी	करते	चिव	चिव		
Prediction after					करते				
		4	1	0	1	0	0		
	किती वेळ लागेल			किती	वेळ	लागेल			
Prediction after				कि	वे	ला			
		3	3	1	1	1			
	जोवरी पैसा तोवरी बैसा			जोवरी	पैसा	तोवरी	बैसा		
Prediction after					पैसा				
		4	1	0	1	0	0		
	तू कसा आहेस			तू	कसा	आहेस			
Prediction after					कसा	आहे			
		3	2	0	1	1			
	राव चढले पंत पडले			राव	चढले	पंत	पडले		
Prediction after									
		4	0	0	0	0	0		
	काळाकाळा कापूस पिंजला रे			काळाकाळा	कापूस	पिंजला	रे		

Prediction after									
		4	0	0	0	0	0		
	भारतभाग्यविधाता								
Prediction after									
		1	0	0					
	मामाची बायको सुगरण रोजरोज पोळी-शिकरण			मामाची	बायको	सुगरण	रोजरोज	शिकरण	
Prediction after									
		6	0	0	0	0	0	0	
	नदीनाल्यांना आला पूर			नदीनाल्यांना	आला	पूर			
Prediction after									
		3	0	0	0	0			
	कामापुरता मामा आणि ताकापुरती आजी			कामापुरता	मामा	आणि	ताकापुरती		
Prediction after						आ			
		5	1	0	0	1	0		
	झोंबे अंगा वारे काया थरथरे			झोंबे	अंगा	वारे	काया		

Table 5 . Truth table of phrases used for evaluation of keyboard with lesser predictions

	झोंबे अंगा वारे काया थरथरे			झोंबे	अंगा	वारे	काया		
Prediction after									
		5	0	0	0	0	0		
	पळस गेलं कोकणात तीन पानं चुकेनात			पळस	गेलं	कोकणात	तीन	चुकेनात	
Prediction after							ती		
		6	1	0	0	0	1		
	खायला कोंडा नि निजेला धोंडा			खायला	कोंडा	नि	निजेला		
Prediction after				खायला					
		5	1	1	0	0	0		
	पावसाच्या रेघांत खेळ खेळू दोघांत			पावसाच्या	रेघांत	खेळ	खेळू		
Prediction after									
		5	0	0	0	0	0		
	सारे भारतीय माझे बांधव आहेत			सारे	भारतीय	माझे	बांधव		
Prediction after					भारती	माझे			
		5	3	0	1	1	0		
	पुढे मला काही कल्पना सुचू लागल्या			पुढे	मला	काही	कल्पना	लागल्या	
Prediction after				पु	मला	काही			
		6	3	1	1	1	0	0	

Table 5 . Truth table of phrases used for evaluation of keyboard with lesser predictions

	परहित आधी नंतर स्वहित साधावे			परहित	आधी	नंतर	स्वहित		
Prediction after									
		5	0	0	0	0	0		
	कराग्रे वसते लक्ष्मी करमध्ये सरस्वती			कराग्रे	वसते	लक्ष्मी	करमध्ये		
Prediction after									
		5	0	0	0	0	0		
	दैव देते आणि कर्म नेते			दैव	देते	आणि	कर्म		
Prediction after					देते	आ			
		5	2	0	1	1	0		
	पचापचा शिव्या देई खाताखाता पान			पचापचा	शिव्या	देई	खाताखाता		
Prediction after									
		5	0	0	0	0	0		

Table 5 . Truth table of phrases used for evaluation of keyboard with lesser predictions