A Comparison of Input Entry Rates in a Multimodal Mobile Application

Aleksi Melto, Markku Turunen, Jaakko Hakulinen, Anssi Kainulainen, Tomi Heimonen

Speech-based and Pervasive Interaction Group, Tampere Unit for Computer Human Interaction, University of Tampere, Finland
firstname.lastname@cs.uta.fi

Abstract

This paper presents results from a comparison of text and speech input methods on TravelMan, a multimodal route guidance application for mobile phones. TravelMan provides public transport information in Finland. The application includes a range of input methods, such as speech and predictive text inputs. In this paper we present results from the user evaluation focusing on entry rates of multi-tap text input, domain-specific predictive text input, and speech input. In addition to objective metrics, we present findings about user experiences related to different input methods.

Index Terms: input entry rate, mobile user interfaces

1. Introduction

Entering information is challenging in small mobile devices because of the limited entry methods. The need for data entry, however, is a major concern due to increasingly advanced applications available for mobile phones. Nowadays even regular phones include the possibility to run custom applications (e.g., MIDlets in Java enabled phones). This has enabled the development of multimodal and distributed mobile applications that take advantage of e.g., speech, graphics, non-speech audio, haptics and positioning information. These new developments can provide more efficient ways for data entry. At the same time, the mobile context of use has brought along new application domains for speech-based and multimodal systems, such as mobile public transport navigation assistants considered in this paper.

There are numerous approaches for predicting the efficiency of different input methods. For example, [5] presents a model for data entry rates for speech recognition and alternative input methods, such as T9. The model suggests that there are several speech-based input methods that can be faster than generic predictive text-inputs and regular multi-tap text inputs. However, in specific domains, such as route guidance and navigation applications considered here, predictive text inputs can be made very efficient with domain knowledge. There have been few comparisons of spoken data entry methods in the mobile context, and speech is often dismissed for very small vocabulary cases [3]. Furthermore, it is crucial to get results from user studies to see what realistic entry rates are, and how different input methods compare when subjective evaluations are considered.

In this paper we report how speech input compares with regular text input (multi-tap) and domain-specific predictive text input in the case of the mobile multimodal route guidance application. In our case, the domain consists of addresses (street names and numbers, e.g., “Mannerheimintie 23”) and points of interests (“main railways station”). The results can be applied for similar domains, i.e., domains where vocabulary is quite large, but well-known and not captured well with generic methods, such as generic predictive text input methods.

In the rest of the paper, we first present the Travelman route guidance application and its evaluation, focusing on the comparison of text and speech inputs. Finally, we present key findings from the subjective evaluation of the application and different input methods. The paper ends with discussion and conclusions.

2. TravelMan Application

TravelMan [6] is a multimodal mobile application providing route guidance information for public transport in Finland. It supports commuter travel with metro, tram, train and bus traffic in cities and long-distance traffic in the rest of the country. There are two main functions: (1) planning a journey and (2) interactive guidance during the journey. In the journey planning phase a user gives the departure and destination addresses or locations using one of the several available input methods, as discussed in the next section. When the user has given the addresses, the system computes a set of applicable routes that can be browsed. After selecting a suitable route, the user can simply listen how the journey progresses, or navigate in the route description manually.

The use of TravelMan is based on multidirectional menus that are operated with the directional keys of the telephone. As illustrated in Figure 1, menus are presented using a reel metaphor: items in a menu are on top of each other and the user can roll the reel to select menu items. The currently selected node is enlarged to provide more information and to ease viewing the information on the small display. As the adjacent items are visible, the user always has a context for the current selection. This interface solution is inspired by fisheye techniques such as Fisheye Menus [1] (and in general, focus and context visualization techniques).

Figure 1: TravelMan user interface. Left: Main menu with the reel interface. Right: Predictive text input using the reel interface.
variations of text input. Speech input is based on using the keypad to start and stop the recording of audio. For speech recognition, the application uses a distributed system where the audio is sent over network to a server-based Finnish language speech recognizer. Since the vocabulary is in the range of thousands of words, server-based recognition is necessary to make the recognition robust and fast enough.

The second way to enter addresses is two variations of text input. In addition to normal multi-tap text entry, a (T9-style) predictive text input, optimized for the address data entry, is provided. The language model for the prediction contains all the valid street and place names supported by the application. The reel interface is used in the predictive entry mode so that the reel contains all valid addresses in the beginning and narrows down the list in real time when the user enters new characters. At any time, the user can select a street name from the reel. When a street name has been selected or completely entered, the predictive text input automatically switches to number mode for entering the street number. Similarly, numbers from 1 to 100 are recognized in speech input.

In the user evaluation a version of Travelman targeted for cities of Helsinki, Espoo and Vantaa was used. In this domain, the language model consisted of 8896 street names and addresses, 2053 place names, and numbers from 1 to 100, totaling 11049 words.

3. User Experiment

The TravelMan application has been in public pilot use since the spring of 2007, accumulating over 1000 real users in the pilot phase. In order to find out the efficiency and user experience of different input methods, we arranged more formalized user evaluations in a lab environment in the autumn of 2007. The results from the lab experiment are presented in the following.

3.1. Test Setup

38 students from the local university participated in the evaluation (27 male, 11 female). Their age ranged from 18 to 45 years, with an average age of 24 years. 21 participants had prior experience with Series 60 smartphones, which were used as the test bed in the evaluation. Both objective and subjective metrics were collected to analyze the interactions and elicit feedback from the participants.

Participants were introduced to the TravelMan application using a web-based wizard. Main features of the TravelMan application were presented, but the actual usage instructions were not revealed at that point. After introduction, users filled a web-form questionnaire to record user expectations and establish an acceptable level and desired levels of service quality for various features.

The test took place within one to two weeks after the introduction and service quality questionnaires. A Nokia N95 smartphone (with a numeric keypad) was used in the test, and data was gathered with the test application’s internal logging system that stored every key press and action. The participants were given four exercise tasks and 21 evaluation tasks. The exercise tasks did not include the use of any input method for addresses (speech, multi-tap text, predictive text); only general navigation to familiarize the participants with the reel user interface. The actual tasks were designed in a way that participants had to enter either the start location or the destination in each task, but the actual answer (for example, the time the next bus departed) was not to be found in the address input view. If the initial attempt to accomplish a task failed (e.g. due to a speech recognition error or mistyped address), the participant was requested to try again. At this time, the participants were not informed that the test was related to input methods, but instead that it was a regular usability test to discover problems in the software.

The evaluation was organized as a within subject study. The three task sets were the same for all participants and the order of modality presentation was counterbalanced. The task set – modality pairings depended on the group. The tasks were always presented in the same order within modality. After completing the task set with the given modality, the participants were asked to fill in a questionnaire. Questions consisted of the exact same statements they were asked in the pre-test questionnaire. This time the participants gave only one value - based on their experience of use.

In order to keep task sets comparable, we selected the addresses in each set based on the minimum button presses that are required to select that particular address with the predictive text input reel. For example, the first address of every set required eight key presses in the optimal case.

The connection speed between the mobile device and the server varied greatly in the experiment. 24 participants used 802.11g WLAN connections, 9 used 3G connections (max. 384 kbit/s, but in practice a lot of variation), and 5 used EGPRS connections. This affected mostly speech tasks, since transferring the audio data to the server took most of the time compared to transferring of the text input strings. For speech, total times for task completion varied from 5 seconds to 237 seconds. Input time and transfer time are reported separately, where appropriate. Furthermore, there were large differences (from 5 seconds to 310 seconds) in predictive text inputs because of long learning times of some users.

Next, we present main results of the evaluation focusing on entry rates. Median values are reported, unless noted otherwise, because of aforementioned extreme values. All results mentioned are either highly significant (p < 0.001) or significant (p < 0.01) by Pearson’s Correlation or ANOVA, unless otherwise noted.

3.2. Results

As seen in Table 1, the median task completion time was 13.7 seconds for speech inputs, 17.2 seconds for predictive text inputs, and 30 seconds for multi-tap text input. Speech recognition rates varied greatly between different users. The range was from 100% to about 45%. Overall recognition rate was 70.4% and 97% of recognition task cases were completed successfully within three attempts.

In the case of speech input, the total entry time was greatly affected by the data transfer time (7.8 seconds). The median time for giving a speech input was 5.7 seconds (SD. 5.4). This includes all retries caused by misrecognitions, in average each speech input was given 1.47 times. For multi-tap and predictive text input, the transfer time is around one second, and the actual input time was 28.8 seconds (SD. 13.7) for multi-tap text input, and 15.6 seconds (SD. 32.3) for predictive text input.

As seen from the results, speech is the fastest input method, even with long transfer times caused by distribution and error prone speech recognition. Theoretically, it could be superior: the time for giving a single entry with speech was 4.6 seconds (SD. 2.2), mean for audio recording length being 3.3 seconds (SD. 1.5). For normal and predictive text inputs, the corresponding times were 28.2 (SD. 16.9) seconds and 14.8 (SD. 31.5) seconds, respectively.
### Table 1: Average input entry times per modality.

<table>
<thead>
<tr>
<th>Modality</th>
<th>Number of tries</th>
<th>Input Time</th>
<th>Transfer time</th>
<th>Total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-tap</td>
<td>1.02</td>
<td>28.8</td>
<td>0.8</td>
<td>30.0</td>
</tr>
<tr>
<td>Predictive</td>
<td>1.05</td>
<td>15.6</td>
<td>1.1</td>
<td>17.2</td>
</tr>
<tr>
<td>Speech</td>
<td>1.47</td>
<td>5.7</td>
<td>7.8</td>
<td>13.7</td>
</tr>
</tbody>
</table>

3.2.1. Learning

A clear learning effect can be seen in the case of predictive text input when comparing input rate efficiency between the tasks. With speech input, number of input task (re)tries was 39% higher in the first tasks (1.9) compared to the rest of the tasks (1.3), the average being 1.5, as seen in Table 2. This difference is significant (One-Way ANOVA, \( p < 0.05 \)). Otherwise there are no differences. As usual with speech recognition, the recognition accuracy varied between the sentences to be recognized. For multi-tap text input, no effect of learning is seen.

Table 2: Entry rates for different modalities.

For predictive text input the average percentage of incorrect entries was 8%, and shows only slight decrease over tasks. However, there are significant differences when input entry efficiency (characters / second) and the number of “clear key” presses are compared. As seen in Table 2, in the first task the input entry rate was 5.35 wpm, and it increased to 11.58 wpm for the last task, average being 10.02 wpm.

Table 3: “Clear key” presses.

When the number of “clear key”-presses are compared, there are clear differences, as seen in Table 3. The amount of “clear key”-presses was surprisingly high in the first task (0.41 / character), but decreased rapidly to a very low amount (0.02 / character). This clearly shows how people learned to use the predictive text input method during the experiment.

As can be seen from the tables, there are only minor differences in the case of multi-tap text inputs. Furthermore, its efficiency is mostly characterized by the text to be typed. For example, some of the tasks included addresses with Finnish characters with umlaut, and therefore required more key presses / characters than others.

When comparing the total times required for tasks within modalities, as illustrated in Figure 2, it can be seen that the total times decrease quite much after the first task, and continued to decrease in most cases. Figure 2 shows the development of total time and it also shows how predictive text input outperformed speech in last two cases. Predictive input was 1.6 seconds faster than speech within last task when examining median values and 13.8 seconds with mean values.

By the final task the difference in total time between predictive and speech within final task was significant. However, without the speech recognition performance issues with handset to server communication, speech input would have been faster, as the median and mean for input were both under seven seconds.

Figure 2: Total input entry times.

3.2.2. Text Input Characteristics

Number of total key presses was much lower in predictive text entry than in multi-tap, as predicted. On average, 38 key presses (SD. 17.9) were needed to get the correct address in multi-tap text entry mode, while in the predictive text mode the number was 11 (SD. 21.2). In predictive text input condition, selections from the reel were common. In 86% of the cases, the address was selected from the reel menu. While maximum number of moves within individual task was 46, only in few cases the list was scrolled further than few items. On average, participants moved 1.8 steps, median and mode being 1. In other words, a typical user typed characters until the correct address was the nearest (in either direction) item in the reel menu. The “clear key” was used to remove characters more often in multi-tap text input than in predictive text input. Percentages of address entries where the clear key was used were 70% and 24%, correspondingly, and average numbers of clear key presses per entry were 2.2 and 1.8.

3.3. Subjective Evaluation

The SERVQUAL-method was used to collect subjective metrics. It estimates what is the current state of the application from the basis of user expectations and experiences. The method produces a subjective measure of the gap between the pre-test user expectations for an acceptable level and a desired level, and the post-test perceived level (i.e. user experience). Originally developed by marketing academics, it has been successfully adapted for evaluation of speech applications [2].

We calculated these measures for different usability dimensions (speed, pleasantness, clearness, error free use, learning curve, naturalness, usefulness, and willingness to use), for each modality (multi-tap text input, predictive text input, and speech input), and the application itself. For example, one statement was “Speech input is quick to use”. When analyzing the results, we found significant differences.

First, we found cases where the perceived quality was better than the desired level, or very near to it. These cases relate to the usefulness and user’s willingness to use predictive text input and the TravelMan application. In other words, users found predictive text input and the TravelMan superior to their expectations. Otherwise, in the case of multi-tap text input and speech input, users found these input
methods quite positive, but ranked the multi-tap text input higher than speech input. When compared to the objective results, we could not find correlations between the objective measures and the reported user experiences, even when actual speech recognition error rates and perceived error rates were considered. The only correlation was between the actual total task times for speech input and the perceived speed of speech input.

Overall, users considered the predictive text input method superior to speech and multi-tap text inputs. Speech input was the least favored method. Slow connection times and erroneous speech recognition could not explain these results, or at least correlations between groups were not found. Further details of the study are presented in [7].

4. Discussion and Conclusions

We have compared speech input to multi-tap and domain optimized predictive text input in a mobile route guidance application. For speech inputs, objective results show that speech is both in theory and practice the most efficient input method, even with relatively high error rates and slow response times. However, people do not have too high expectations for speech inputs, and although user experiences are reasonably good compared to the expectations, predictive text input and even multi-tap text input outperform speech in subjective user ratings.

Speech recognition rates were on levels where they can be considered problematic, if compared to the typical 99% rates seen in advertisements of commercial products and reports of speech recognition developers. However, our results are more comparable to the error rates people are getting in real use. As usual, the rates were different for different users, some having clearly unusable rates, while some users experienced perfect recognition. In some cases, the speed advantage that spoken input can have is lost in the error correction effort. Street and location names are usually longer words, which is good for speech recognition. On the other hand, most addresses end with the word “street” or “road”.

Lengthy response times were problematic for speech input. These were caused by the distributed speech recognition and slow data transfer connections. Despite these problems, the average total task completion time was shorter than in the case of predictive text input and superior compared to multi-tap text input. This can be seen in the user attitudes towards speech inputs, which were quite high, matching the required quality. With the given recognition rates and absolute error percentages, we would have anticipated more criticism.

The multi-tap text input method was clearly slower than the alternatives in the experiment. When we compare the predictive and multi-tap text entry, the difference in average number of key presses is very high. The difference in speed, while high, is not relatively as great. This signifies that while users mostly focused on entering characters in traditional text entry, they used more time in other tasks, presumably in visual search within the reel list, in predictive text entry. The widely used “select from list” approach supports this and the positive subjective ratings towards the predictive text entry suggest that the participants like this form of data entry. The results show that domain specific predictive text inputs can be a very efficient method for similar domains. When the number of required key presses increases, the selection from a list becomes more appealing.

On subjective evaluations, the predictive text entry was very well received; it surpassed the expectations gathered beforehand. This is exemplified by some participants, who did not use T9 normally, having difficulties to understand the logic of predictive text inputs in the beginning. This is manifested as shuffling through address lists, and several attempts to input text in the manner of multi-tap text input. Statistically, it can be seen as an exceptionally high amount of key presses, and long performance times that go up to minutes. However, in many cases, when they finally understood the method, it was considered with enthusiasm, as indicated in the subjective ratings and informal feedback (Laura: “I didn’t like predictive input before, but once you got to know with it there should be no regular text input at all!”).

Overall, the results show that speech inputs are theoretically superior to text inputs, and they outperform other methods in practice as well, even with relatively high error rates and slow response times. On the other hand, domain specific predictive text entry can be a very efficient entry method, and it had significant learning taking place. In the end, it was faster than speech in the last two tasks, mainly because of slow connections.

The use of speech inputs is likely to be dependent on the added value they provide, since predictive text inputs, and even multi-tap text inputs, outperformed them in subjective user opinions, and predictive text input is hard to beat without a local recognizer or fast connections, and high accuracy rates. However, on mobile settings, speech inputs can give clear advantages, since pen-based soft-key text input has been shown to be slower while walking [4], and it is likely to be true for the predictive text inputs. We will conduct user experiments when people are on the move, and compare the efficiency of predictive text input and speech input.

5. Acknowledgements

This work was supported by the Technology Development Agency of Finland (FENIX-programme, PUMS-project). We thank Ville Antila for his contribution in the user tests.

6. References