

# Query Suggestions for Mobile Search: Understanding Usage Patterns

Maryam Kamvar

Google Inc / Columbia University

1600 Amphitheatre Parkway, Mountain View, CA  
[mkamvar@google.com](mailto:mkamvar@google.com)

Shumeet Baluja

Google Inc

1600 Amphitheatre Parkway, Mountain View, CA  
[shumeet@google.com](mailto:shumeet@google.com)

## ABSTRACT

Entering search terms on mobile phones is a time consuming and cumbersome task. In this paper, we explore the usage patterns of query entry interfaces that display suggestions. Our primary goal is to build a usage model of query suggestions in order to provide user interface guidelines for mobile text prediction interfaces. We find that users who were asked to enter queries on a search interface with query suggestions rated their workload lower and their enjoyment higher. They also saved, on average, approximately half of the key presses compared to users who were not shown suggestions, despite no associated decrease in time to enter a query. Surprisingly, users also accepted suggestions when the process of doing so resulted in an increase in the number of total key presses.

## Author Keywords

Mobile search, text entry, query prediction.

## ACM Classification Keywords

H5.2. Information interfaces and presentation: User Interfaces.

## INTRODUCTION

Typing text on a standard 9-key cell phone is difficult and time consuming. The average query on Google's mobile search page is 15 letters long, but takes 30 key presses and approximately 40 seconds to enter [1]. To address these inefficiencies, a variety of text prediction techniques have been proposed, including eZiType, iTap and T9. We consider systems such as eZiType to be word *suggestion* systems, since they complete the word before all the letters are pressed. Systems such as T9 (in its common instantiation) are word *disambiguation* systems that use a dictionary to map 9-keypad button presses to words. A more detailed analysis can be found in [2,6].

In this paper, we study usage patterns of a query suggestion system. We show that mobile phone users will rely heavily on suggestions if they are provided. Users who were asked to enter queries on a search interface with query

suggestions rated their workload lower, their enjoyment higher, and saved nearly half of the key presses than the users who were not shown suggestions.

Our primary goal is to build a usage model of query suggestions in order to provide UI guidelines for mobile text prediction interfaces. We show that users accept the correct suggestion by the third time it is shown. This behavior implies that it is safe to replace suggestions after three appearances in the list, which allows us to show more suggestions using less screen space. Surprisingly, users will often accept suggestions even if the process of doing so results in an increase in the number of total key presses.

## EXPERIMENTS

Each user was given a phone with an instrumented Java (J2ME) application (Figure 1). At the start of the study, users were given a verbal outline of the user study; that they would be entering 23 queries on a mobile phone and then would be asked a series of questions regarding their own preferences and experience. They were advised to commit the query to memory when it was presented, as it would not be displayed on the query input screen. Users were informed of the position of the *OK* key and of the existence of the *remind me* key (which they could use in the case they forgot the query). No mention was made of the query suggestions, or of any other interface details. Aside from this verbal introduction, the study was not moderated.

The experiment consisted of 2 phases: a query-entry phase and an evaluation phase. In the query-entry phase, users were shown 2 types of screens: a *query display* screen (Figure 1a) and a *query input* screen (Figure 1b). Users progressed from one screen to the next by pressing the *OK* key (Figure 1a). The *query display* screen informed the user of the query to enter. The *query input* screen consisted of a text box; only multi-tap text entry was enabled in this text box. If the user mis-entered a query, the application would display an error screen, which contained the correct query (Figure 1c). From the error screen, the user was redirected to the previous *query input* screen. In this phase, the exact key press sequence and associated times were logged.

In the second phase, the user was presented with the NASA-Task Load Index (TLX) [3] scales and comparison questions (Figure 1 d,e). In addition to the NASA-TLX scales, users rated the "enjoyment" of the experience. This rating was not used in the workload calculation.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2008, April 5–10, 2008, Florence, Italy.

Copyright 2008 ACM 978-1-60558-011-1/08/04...\$5.00

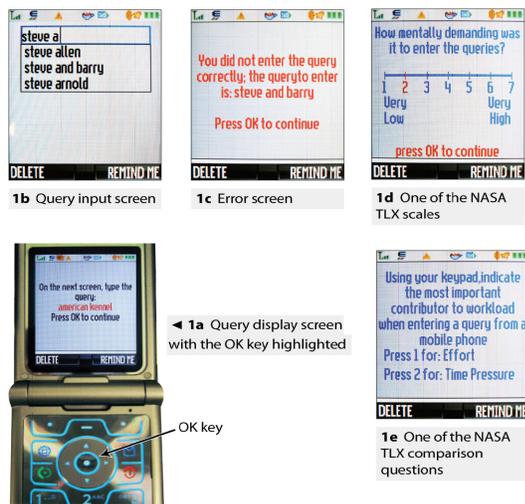


Figure 1: Snapshot of application provided to each user.

### Interface Variants Tested

Six interfaces were studied; each user was assigned a single interface for the duration of the study. The six interfaces differed only in the number of suggestions displayed as the user was typing the query. The number of suggestions ranged from zero (no suggestions) to a maximum of five suggestions (the maximum number of suggestions that fit on the screen without requiring the user to scroll).

As shown in Figure 1, the suggestions (if present) were presented in a drop-down list below the textbox. Users could access the suggestions by pressing the down key. The first down key press would remove the cursor from the textbox and highlight the first suggestion. Each subsequent down key press would highlight the next suggestion in the list. Users could accept a highlighted suggestion by pressing the *OK* key and the contents of the textbox would be replaced with the suggestion. Text entry was disabled while traversing the suggestions list; however, a user could scroll up past the first suggestion to re-enable text entry.

### Queries & Suggestions

The 23 queries that users were required to input were chosen from the Google query logs and fulfilled the following 4 constraints. Each query consisted of only letters a-z and spaces, was 15-16 letters in length (including spaces), required 30-31 key presses (assuming multi-tap input, no use of suggestions, and no errors), and had two sets of consecutive letters that appeared on the same key on a 9-key keypad (Figure 1). The length of each query and number of key presses required for each query were chosen to be consistent with average length of mobile queries and the average number of key presses needed to enter them [4]. Three sample queries are: “arlight cinema”, “the little door”, and “american racing”.

The queries were presented in the same order for all users. For queries numbered 7,11,15,19 and 23, the correct suggestions were “hard-coded” to appear after the user

typed the third letter in the query, and the correct suggestions appeared at positions 1,4,3,2 and 5 respectively. With each additional letter, the correct suggestion moved up one position, until it reached the top position. For all other queries, their suggestions followed the convention of Google Suggest [5]. The suggestions appeared in decreasing query frequency order, as found in the Google query logs. The suggestions only had upward mobility as the prefix expanded.

### Users & Datasets

Thirty users who owned Motorola RAZRs participated in the study; a single phone type was used to eliminate confounding factors. The users consisted of 11 engineers, 5 sales reps, 6 product managers, and 8 employees from other departments. No users were chosen from groups working with mobile products. The 30 users were divided into 6 groups, and each group was assigned one of the interface variations. To control for text entry expertise across variations, each group consisted of 3 expert users (those who reported sending an SMS at least 1/day), 1 regular user (SMS: 1/week) and 1 novice user (SMS: 1/month).

We analyzed the 125 trials with “hard-coded” suggestions (from the 25 users who saw suggestions) to study how movement affects acceptance patterns. The remaining 450 trials were considered for time and key presses analysis. When evaluating average time to enter a query, we disregarded the 44 queries where a user either used the *remind me* key, or entered the query incorrectly (and was shown the error screen). Of these 450 queries entered on interfaces which displayed the drop down list of suggestions, 435 had a useful suggestion shown before the user finished entering the query. We consider useful suggestions to be partial completions (where the suggestion completes part of the desired query), super completions (where the suggestion is a superset of the desired query) or full completions (where the suggestion is the desired query). The distribution of useful suggestions was weighted towards full completions; 348 queries were shown with full completions in their suggestions list (Showing a full completion was not exclusive of showing partial or super completions). Unless otherwise noted, the statistics in this paper refer only to full completions. The histogram of number of letters a user typed before the full completion appeared in the suggestions list is shown in Figure 2.

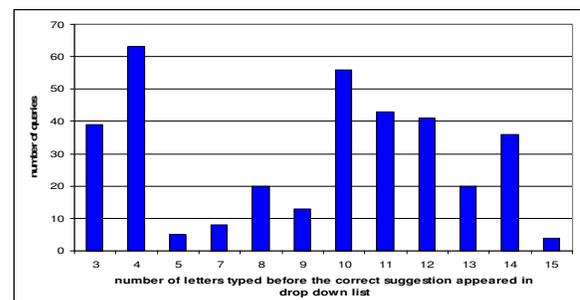


Figure 2: Histogram of the number of letters entered before the full query completion appears in the suggestions list.

## FINDINGS

Users who were asked to enter queries on a search interface with query suggestions rated their workload lower, their enjoyment higher and saved nearly half of the key presses than the users who were not shown suggestions (Table 1). Surprisingly, time to enter a query was *not* reduced with the decreased number of key presses.

**Table 1: Workload, enjoyment, key presses and time per query**  
(Note that the number of key presses for interfaces with suggestions include those required to select and accept a suggestion.)

	<i>no suggestions</i>	<i>with suggestions</i>	$p \leq 0.05$
workload	26.4	20.5	Yes
enjoyment	1.8	3.2	Yes
# key presses	31.2	18.7	Yes
time (s)	20.1	20.8	No

Enjoyment and overall workload of the task reveal the user's qualitative perception of query entry. On average, users who were not shown any suggestions ranked the enjoyment of entering the 23 queries as a 1.8 on a scale from 1 to 7 (max=3, min=1). Users who were shown suggestions rated their enjoyment at an average of 3.2 on the 7 point scale (max=7, min=1).

We used NASA-TLX to determine the user's perceived workload, as per the conventional formula [3] and found that a user's perceived workload reduced over five points when the user was presented with suggestions.

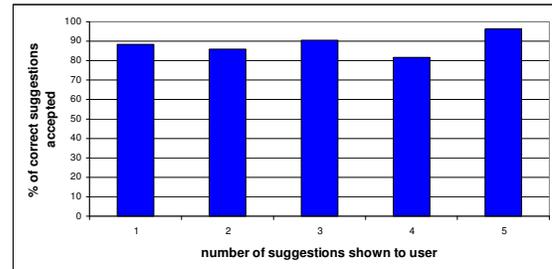
The number of key presses and amount of time needed to enter a query reveal quantitative efficiency metrics for the task. The number of key presses needed to enter a query nearly halved for users who were given suggestions. However, the time to enter a query did not reduce with the decrease in key presses. This trend indicates that the presence of query suggestions may *slow* the number of key presses/second. This is preliminary evidence that displaying suggestions trades off an easier input experience with an increase in cognitive load; this tradeoff has also been noted in studies of other mobile text entry interfaces which display suggestions [7]. For more evidence of the cognitive load introduced by displaying suggestion to the user, we looked at the 27 queries where none of the displayed suggestions were accepted by the user (these instances were spread across 11 expert, 3 average and 1 novice user). The average time to enter these queries was 30.3 seconds. This is significantly longer than the average of 20.1 seconds it took users to enter queries without suggestions, which is a strong indication that users are spending a significant fraction of their query entry time processing the suggestions.

### Users rely heavily on query suggestions

100% of the users who were shown suggestions accepted at least one suggestion; the average number of accepted suggestions (including full, partial and super completions) per query was 0.9. If the correct suggestion appears in the query suggestion list, users will scroll down and accept it rather than complete the query by typing an overwhelming

majority of the time. Across all interfaces, users accepted a correct suggestion 88.5% of the time. This was computed from a sample of 348 queries for which the complete suggestion appeared in the drop down list of suggestions before the user finished entering the query.

The number of suggestions shown to the user did not impact the high acceptance rate. Figure 3 shows the acceptance rate for each interface; the differences across the interfaces are not statistically significant.



**Figure 3: Percentage of times the correct suggestion is accepted.**

### Users often ignore the cost of accepting a suggestion

The most common method of judging the cost-benefit of using a suggestion is by the number of key presses saved. However, users did not make that cost-benefit analysis when considering whether to accept a listed suggestion. Of suggestions that were presented to the user when the number of key presses left to type was less than the number of down and enter key presses required to select the suggestion, 50% were accepted. In these cases, accepting a suggestion resulted in a net *increase* in key presses over completing the query by typing. It is likely that users are not making this tradeoff because it is fairly complex; users would need to continually evaluate the number of letters left, how many key presses each remaining letters entail (most letters require multiple key presses) and the position of the correct suggestion (the number of downs).

We also examined a simpler model for evaluating the cost-benefit of accepting a suggestion: we look at the number of letters left to type versus the position of the suggestion in the list. We find that users commonly do not consider this simpler model either. Of the suggestions which were presented to the user when the letters left to type were less than or equal to the position of the suggestion, 73.1% were accepted. In the majority of the cases, users do not engage in a cost-benefit analysis when deciding to accept query suggestions, neither on the key press level nor on the letter level.

### Users will accept a correct suggestion quickly

If a user has not accepted a suggestion after it is shown three times (*i.e.* for three unique prefixes) it can be taken as a strong signal that the suggestion is not the user's intended query. Correct suggestions were shown an average of 1.4 times before they were accepted. 97.4% of accepted queries were selected from the list by the third time they were shown. Figure 4 shows the histogram of number of times a suggestion was shown before it was accepted.

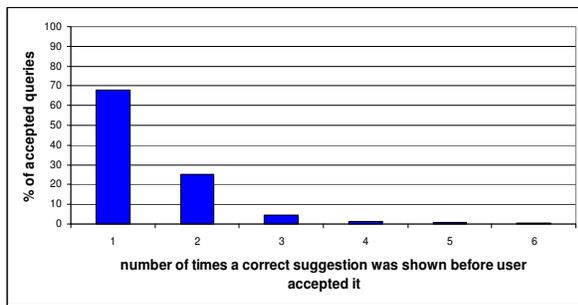


Figure 4: Number of times a suggestion is shown before acceptance.

Showing more suggestions may hinder the efficient usage of suggestions. Users who are shown fewer suggestions are likely to accept a suggestion earlier, perhaps because with fewer suggestions it is easier to identify a correct suggestion. Figure 5 shows the cumulative percentage for the number of times a correct query was shown before it was accepted. We see that the median shifts towards an increasing number of appearances as the number of suggestion shown increases.

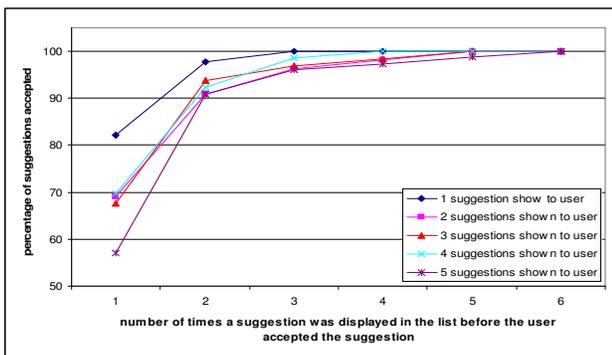


Figure 5: The cumulative percentage of number of times a correct suggestion was shown before it was accepted.

Another factor which impacts how quickly a user will accept the suggestion is the movement of the suggestion in the list. Suggestions in this model (and all models in which the suggestions are ordered by the frequency of occurrence) move upwards in the list, though their exact movement may not be rigidly defined. We looked at two cases: the case where the suggestion moved one position up with each new letter and the case where the suggestion stayed in the same position after a new letter was entered. Counter intuitively, movement of suggestions in the list seems to hinder efficient acceptance as suggestions which moved were accepted later than suggestions which stayed stationary. To measure this, we looked at queries numbered 7,11,15,19 and 23 which, as previously mentioned, had their full completions hard coded to appear after the user typed the third letter at positions 1,4,3,2 and 5 respectively. For these queries, the full suggestion moved up one position with each additional letter until it reached the top position. From that set, we disregarded the non-moving suggestions. (shown once before acceptance, etc). The average number of times a suggestion was shown before the user accepted it was 3.8, and the average number of positions these

suggestions ever occupied was 2.5. For the suggestions which moved from their initial position, but not necessarily in a predictable linear manner, the average number of times they were shown before accepted was 2.5. These suggestions occupied an average of 2.0 positions. To compute the average number of times a “stationary” suggestion was shown before accepted, we looked at all suggestions shown more than once whose initial position and position of acceptance was the same. The average number of times a stationary suggestion was shown before it was accepted decreased to 2.2. This trend indicates that the more a suggestion moves in the list, the longer it will take for a user to accept the suggestion.

## CONCLUSION AND FUTURE WORK

Although many text prediction systems exist, the study of text prediction interfaces is largely unexplored. In this paper, we study the usage patterns of query prediction interfaces to guide UI design and to provide metrics to better estimate the *realized* performance of a query prediction models. When designing a UI for query prediction, the overarching guideline is to show as many suggestions in as small of a list as possible. Keeping the length of the list constant, we can maximize the number of suggestions shown to the user in two ways. The first involves replacing the suggestions viewed three times, because we have found that if a suggestion is not accepted by the third time it is displayed in the list, it is unlikely to be the correct suggestion. The second, when optimizing for efficiency, is replacing suggestions that create a net increase in key presses if accepted; users are unlikely to compute the cost-benefit analysis of accepting a suggestion.

We have studied the aggregate effects of showing suggestions to the user. A more granular study may also be interesting; for example, to determine when there is an inflection point in the number of suggestions shown, where the suggestions actually hinder performance and reduce satisfaction. In this study, we employed cell phones with 9-key keypads. It will be interesting to determine the effects of different search mediums: do users with miniature QWERTY keyboards rely on suggestions less frequently, and how does it compare to conventional computers?

## ACKNOWLEDGMENTS

We would like to thank Robin Jeffries, Bay Wei Chang, Andrea Knight, Helena Roerber, Scott Jenson & Sanjay Mavinkurve for their feedback.

## REFERENCES

1. Kamvar, M., Baluja, S. Deciphering Trends in Mobile Search. *Computer*. vol.40, no.8. IEEE (Aug 2007), 58-62.
2. MacKenzie, I.S., Kober, H., Smith, D., Skepner, E. LetterWise: prefix-based disambiguation for mobile text input. *Proc. UIST 2001*.
3. Hart, S.G., Staveland, L.E. Development of NASA-TLX Results of empirical and theoretical research. *Human Mental Workload*. (1988)
4. Kamvar, M., Baluja, S. A Large Scale Study of Wireless Search Patterns. *Proc. CHI 2006*, ACM (2006), 701-709.
5. Google Suggest. <http://labs.google.com/suggest>.
6. Garay-Vitoria, N., Abascal, J. Text Prediction Systems: a survey. *Univ Access Information Society* vol 4. Springer-Verlag (2006), 188-20
7. MacKenzie, I. S., Chen, J., & Oniszczak, A. Unipad: Single-stroke text entry with language-based acceleration. *NordiCHI 2006*, 78-85.

