12-2011

# PowerSearch: Augmenting mobile phone search through personalization

Xiangyu Liu
*University of Arkansas, Fayetteville*

# PowerSearch: Augmenting mobile phone search through personalization

**Submit By: Xiangyu Liu**

**Submit To: Honors College**

**Mentor: Dr. Nilanjan Banerjee**

**Date: 05/11/2011**

**Abstract**

Cell phone has become a fundamental element of people's life. People use it to call each other, browse websites, send text messages, etc. Among all the functionalities, the most important and frequently used is the search functionality. Based on ComScore, in July 2008, Google was estimated to host 235 millions searches per day. However, unlike the search on desktop, the search on cell phone has one critical constrain: battery. Cell phone performing a normal Google search, the battery drains very fast. The reason is that when sending a query to and fetching the results from Google, cell phone keeps communicating to the website through networks such as WiFi and 3G. Yet, due to the limited bandwidth of the network and the large amount of the results, the time of communication will be very long. As a result, the battery dies very quickly. In order to prevent this fast drain of battery, a new program is proposed to personalize the search criteria and fetch the most precise and personalized results, instead of all the results, from the web. Because only a few results are fetched, cell phone will not be communicating with the Internet. Hence, the battery will not die very fast.

The program can increase the energy-efficiency of the battery and, thus, lengthen the running time of the cell phone.

**Introduction**

Mobile phones are probably the most ubiquitously used devices today, with billions of users. From a simple communication device, cell phones have evolved to a powerful portable mini-computer. Individuals can run hundreds of applications such as web browsing, audio playback, and even playing video on cellular phones. Overall, almost everything that a user can utilize a mobile phone for can be broadly termed as "search". This includes web searching, searching popular portals such as Google or MSN, searching for videos on YouTube, or searching landmarks and destinations of interest using a map application. In addition, due to the portability, cell phones can allow user to do searches at any time, any place, as long that there are available network connections.

While mobile phone systems have evolved both in processing power and portability, there are several challenges that make search on mobile phones difficult. At first, the search engines neglect personalities. That is when different users submit the same query, they will receive the same result, regardless of their different interests. Hence, when hundreds of results are retrieved, only the first several web pages can be displayed, limited by bandwidth and the form factor of cell phones, i.e. screen size. Since the results are monotonous, most users have to spend a lot of time in looking into the next page until they find the most desirable result. Thereby, this search application will continuously uses the WiFi or 3G network, of which the radios on mobile phones have limited bandwidth. As a result, the battery will be exhausted quickly and potentially frustrating the user. Consider the following scenario: a query "cat" is submitted through mobile phones by four different people, consisted of an animal rightist, a biology student, a housewife, and a music fan. Any search engine will give all these people the same results. However, in fact, the animal rightist is looking for the recent news related to cat abuse. Housewife wants to see the best place where she can purchase a cat. The student needs to study all the organs a cat has. Yet, the music fan is probably looking for new Pussycat's new album. Since the results shown to everyone is the same, assume that the results happened to show the most desirable results for the animal rightist. Consequently, all the other three people have to look into the next page to find out the best result for them. Thus, these three people face the problems brought by lack of personalization on the search, as stated earlier.

The above challenges present an interesting case for developing a system that facilitates accurate search while mitigating power, bandwidth, form factor, and monotonous results concern. Stated otherwise, the research problem is to design a scheme that is energy efficient, efficiently uses network bandwidth, and displays only the most relevant results on the small mobile phone screen. A plausible technique to address the problem is to design an intelligent query augmenter that modifies a search query such that it is more precise than what the user typed.

We plan to use techniques from personalized search to develop the above system. The primary insight is to augment a search query through contextual information that is specific to a user. This contextual information is akin to a user and is part of an automatically generated user profile that our system

will generate. This contextual information could be a user's location, past search queries, gender, calendar events, web browsing histories, and age. To illustrate how the system would function, consider the previous example where four people search for "cat". We assume that they did many searches and browsed many websites on their own cell phone. The reason is that if this is the first time a user uses his/her mobile phone, there is no contextual information about this user at all. Hence, there is no way to personalize the search for the first-time user. If these four users used the cell phone for a while, from the individual contextual information, we can determine what this person's interest is. For instance, the music fan probably browsed many music websites, while the animal rightist read a lot of articles about animal abuse. Based on personal contextual information, the query "cat" can be modified accordingly. It can become "cat abuse" for the animal rightist, "cat biology organ" for the biology student, "where to buy cat" for the housewife, and "cat music/pussycat" for the music fan. These modified queries will then be sent to the server. The most relevant results will be fetched and displayed. By doing so, we can solve the problems brought by the inaccurate results and save time and energy.

**Background and Related Work**

Our research builds on previous work on search, personalization, and context aware mobile phone usage. A usability study similar to what we are planning was performed by Yahoo! Mobile to analyze search patterns among users. The study randomly sampled 20 million queries from the U.S. and another 20 million queries from other countries. The result of the study showed that **personal entertainment** was the most popular query [7]. Assuming that this observation is true, personalizing search can be highly beneficial to mobile users.

There are several studies on building applications, which aim at improving an individual's experience while using a cellular phone. A subset of these applications use location data and location-based blogs to improve tour-guiding [4], utilize community-based similarity techniques to build behavioral models for users---this information can be used to tailor search results [5], or create a thesaurus specific to a user [8]. Several other research efforts focus on personal interest, activities, and historical queries to improve search [10, 11].

Contrary to the above research attempts, our system, PowerSearch, is a general search personalization framework. PowerSearch utilizes personal context, including locations, web histories, and calendar event, to create a personal dictionary that can define a user's interests and context. It also uses rule-mining algorithms to determine an appropriate search augmentation that minimizes energy and bandwidth utilization on mobile phones [6, 9].

**Our Proposal**

Someone may ask: in order to personalize the search, why can't the user just simply submit a complete sentence to query the server instead of using contextual information? There are two reasons. First, for a small touch screen a mobile phone has, it takes a long time for users to type in the query. Plus, user does not want to type such a long Second, which is the main reason, computers cannot think in the way that human beings think. Human's brain, consisted of around one hundred billions of neurons, is trained years after years. Computer does not have the computability or the ability to interpret as human's brain. Meanwhile, computer is not trained as long as human's brain is. Hence, when a sentence "what pill should I take if my head hurts", which is very clear to people, is queried, computer may not give the right result, because a search engine is a program that matches the words you give to pages on the web [12].

Consequently, to obtain the most accurate results, user shall use as few key words as possible, instead of sentences. With the the-fewer-word-the-better premise, the contextual information becomes extremely significant and powerful to augment the search, because it can narrow down the range of possible results for the user.

The ideal augmentation is to find out the dependency between the contextual information and the query. Therefore, based on the dependencies, the query can always be specified, and the results will always be the most accurate and desirable. However, the dependent relationship between the contextual information and the query is impossible to find out. The reason is stated as following: If Y depends on X, it means if Y occurs, X has occurred. Yet, if X occurs before Y,

claiming that Y depends on X is bogus, since it is possible that X and Y occur coincidently. Therefore, in reality, there is no way to differentiate that there is a dependency relationship between X and Y and that X and Y occur together coincidently. Hence, even though using dependency can perfectly augment the query, it is impossible to find out the dependency relation.

Therefore, we propose another way of utilizing contextual information to augment query, which is using the correlation instead of dependency. Correlation works as follows: in a set of events, event X happens N times. Out of this N times, event Y happened $K_y$ times. Therefore, the correlation of X and Y is:

**Equation 1**

$$Y \Rightarrow X, at\ Prob(\frac{K_y}{N})$$

Nevertheless, some other events could also occur when X occurs. Therefore, by applying Equation 1to all the events that occur with X, we can obtain a set of correlation as shown in Figure 1.

$$A \Rightarrow X, at\ Prob\left(\frac{K_a}{N}\right)$$

$$B \Rightarrow X, at\ Prob\left(\frac{K_b}{N}\right)$$

……

$$C \Rightarrow X, at\ Prob\left(\frac{K_c}{N}\right)$$

**Figure 1**

Same technique can be applied to all the events. Consequently, we will achieve a list of events associated with their correlated events. Based on the highest probabilities, we can determine how to augment the query, when one of the events occurs.

By applying this technique, our proposed system comprises of two salient components, as shown in Figure 2. First, we design user-oriented profiles or models. These models use history of location visited by the user, calendar events, and search and browsing history to determine appropriate correlations between queries and their semantic meaning. Second, we adopt rule-mining techniques from the computer networking literature [6] to automatically infer high probability rules. The rule-mining technique is another form of the correlation technique, but involves more than two events. The rule-mining technique will be discussed in the later section.

Before inferring the rules, a framework needs to be built to complete the following two tasks. First, retrieve user's information, including web histories, calendar events, etc. Second, analyze this information and establish user's profile, which means to determine what are this user's personality, interests, hobbies, etc.

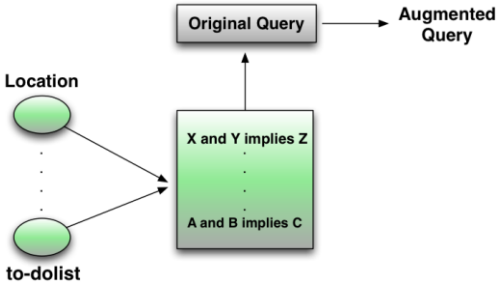


**Figure 2**

**User Information Retrieval**

User information contains contacts, web histories, and calendar events, etc. The reason to retrieve this information is that in cell phone, user's profile is not stored explicitly, because no user has such intention to tell the phone that what his/her occupation and personality are. However, the web histories, bookmarks, and calendar events implicitly reveals this user's personality.

**Web History:**

Web history is one of the most important sources that can easily derive user's information and personality. By browsing web pages, users exhibit their personal interests more explicitly, because they often browse the site they are interested. The relationship between the probability and the interests are shown in Figure 3.

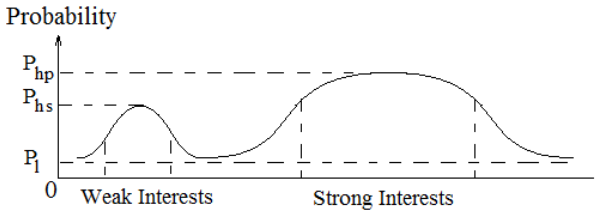


**Figure 3:**

All the users' interests can be categorized into strong interests and weak interests. Strong interests are normally people's habits, hobbies, and personalities. They always have a probability, $P_{hp}$, which means primary high probability. Weak interests, also called temporary interests, are instances that occur occasionally. They have a probability $P_{hs}$, which denotes secondary high probability. If a user always browses websites about basketball, it implies that basketball bonds with this user strongly, i.e. basketball is one of this user's strong interests. Hence, when searching for a good related to sports, this user is, with a very high probability, looking for this good related to basketball. Based on the strong interests, the query can be specified. However, we do not exclude the temporary interests a user might have. For instance, a basketball fan wants to buy a pair of badminton rackets for the first time, which falls into the weak interests category. When he/she queried "badminton rackets", the query should not be augmented into "basketball badminton rackets", which would be meaningless. However, when this user searches for "badminton rackets" for a second time, the query can then be specified into "badminton rackets" with a store name, which he visited last time.

**Calendar Events:**

Calendar events have strong effects at a short period of time on augmenting the query. A calendar event often consists of the name of the event, the location of the event, the happen time, and the recurrent time. When the current time lies within a period before a calendar event time, this event can influence the query with a high probability. When the current time is larger than the event time, this event would have a zero probability on affecting the query, as shown in Figure 4.

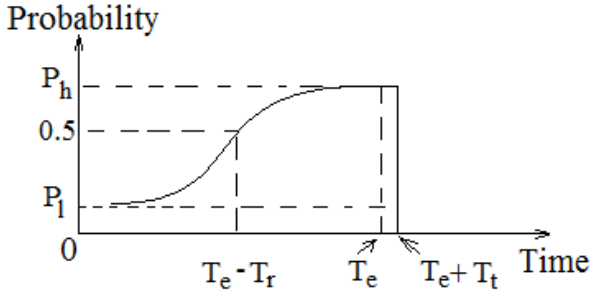


**Figure 4**

$T_e$ denotes the event occurring time. $T_r$ denotes the time range when the probability of the effect on the query is higher than 0.5. When the current time is before ($T_e$ - $T_r$ ), the probability of this event's effect on the query is $P_l$ (low probability). As the time approaches the event date, the probability increases, which means that it is more and more accurate to augment the query by using this event. $T_t$ denotes the tolerance time that how long $P_h$ will still hold after the calendar event. After ($T_e$ + $T_t$ ), the probability drops to zero sharply.

To illustrate the calendar event effect, consider the following example. A woman created a recurrent event on November 5th, which is her son's birthday. Assume that $T_e$ is November 5th.$T_r$ is one week. $T_t$ is one day. If a woman searched for "gift" one week prior to November 5th, i.e. before ($T_e$ - $T_r$ ), it is not likely that she is looking for a present for her son. As time passes by, she might search for "gift" again within one week prior to November 5th. Hence, we can confidently conclude that she is probably looking for a present for her son. The high probability holds for one day ($T_t$) after her son's birthday, because this woman probably forgets about it and still wants to search for a gift for her son. However, after ($T_e$ + $T_t$ ), the probability will drop to zero because the event has passed and will not affect the query any more.

**Location:**

Location is also part of the most significant contextual information. Given User's location, the query can be narrowed down into an area that is closed to the user.

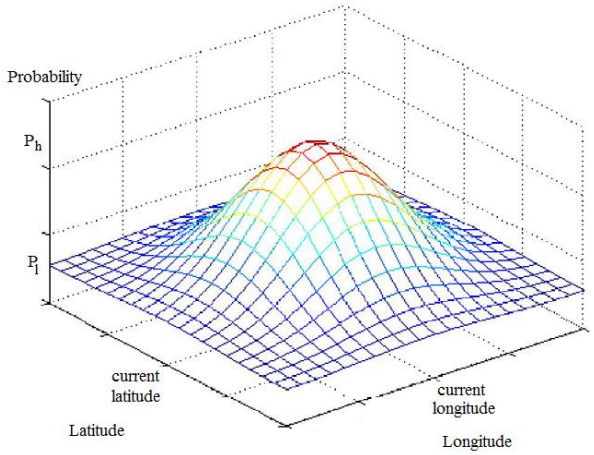


**Figure 5**

As shown in Figure 5, when people do a search at their current location, the probability that the desirable results are close to them is high. As the place becomes further and further away from users' current position, the probability becomes lower and lower. For example, a user searched for "Chinese restaurant". He is most likely searching for the restaurant that is closest to his current location. Therefore, the nearest restaurant should have a highest probability.

**Word Correlation Rules Mining**

Admittedly, queries are all sentences. For a computer, sentences are very difficult to understand. "I'm a student." is different from "I am a student!" to a computer. Therefore, in order to make the word correlation rules more accurate, the queries must be modified so that the computer can understand. For simplification and generalization purposes, we change all the letters into lower case and trim off all the collective nouns, prepositions (i.e. "to", "for", "on", etc.), the verb "to be" (i.e. "is", "am", "'re", "'m". etc.), numbers, and special characters (i.e. "@", "%", "$", etc.). Hence, the left words in the sentence can be considered as significant words. For instance, "There're three students" will be modified into "there students", because "'re" is the short form of "be" and "three" is a collective noun. "I am a student." will be changed into "I student".

The modification is applied to every query user inputted. For all the modified queries, we treat each word in the new query as an event and the whole query as a result. We apply the correlation algorithm to the queries to find out how often a word X has been searched and what the following significant word of X, which is most often co-occurred word with X. For example, a user has queried "where to buy a basketball", "where to buy a baseball", and "where to buy a football". After the modification, the queries become "where buy basketball", "where buy baseball", and "where buy football". Out of these three queries, event "where" happened three times. The following word of "where" is only "buy", which means the most often co-occurred following significant word of "where" is "buy". Similarly, event "buy" also happened three times. However, the following words of "buy" are "basketball", "baseball", and "football". These three words all happened only once. Therefore, all of them are the most often co-occurred following significant word of "buy".

One question might rise: why do we want to mine out the word correlation? The reason is that next time when user queries "where basketball", we can see that word "buy" is a connection word between "where" and "basketball", because "buy" is the most often co-occurred following significant word of "where", and "basketball" is the most often co-occurred following significant word of "buy". Therefore, we can augment the query into "where buy basketball".

After the correlation among the words has been mined, we can now use the rule-mining algorithm to mine out the query-result relationships.

**Rule-Mining Algorithm**

As stated earlier, the rule-mining technique is another form of correlation technique with more than two events. Rule-mining algorithm mines out the rules that would form the building blocks of a grammar, which would constitute a user-specific model. We illustrate our rule mining techniques using an example. Suppose event Q and event X co-occurred and result Z is chosen by user out of all the results. Our rule-mining engine treats the occurrence of event Q and event X as being correlated to event Z. This grammar rule (X and Q implies Z) would occur with a certain probability p that is determined by the number of times X and Q co-occurred with Z. Assuming event Z happened N times. Out of this N time, event X happened $K_x$ times; and event Q happened $K_q$ times. Hypothesize that event X and event Q are independent. Therefore the probability that Z would happen when X and Q co-occur is $\left[\left(\frac{K_x}{N}\right)\left(\frac{K_q}{N}\right)\right]$. Using Equation 2 to express this as following:

**Equation 2**

$$X \wedge Q \Rightarrow Z, at\ Prob\left[\left(\frac{K_x}{N}\right)\left(\frac{K_q}{N}\right)\right]$$

This equation can be expanded based on how many contextual elements are concerned, as shown in Equation 3. $A_1$, $A_2$, …, and $A_m$ are all the events that co-occurred when result Z is returned.

**Equation 3**

$$A_1 \wedge A_2 \wedge \ \ldots \wedge A_m \Rightarrow Z$$
$$at\ Prob\left[\left(\frac{K_{A_1}}{N}\right)\left(\frac{K_{A_2}}{N}\right)\ldots\left(\frac{K_{A_m}}{N}\right)\right]$$

## Query-Result Relation Rules Mining

We add the contextual information, i.e. location and calendar event, into each query-result pair. Thereby, for each pair, we have the following equation:

$$A_q \wedge A_l \wedge A_c \Rightarrow Z$$
$$at\ Prob\left[\left(\frac{K_{A_q}}{N}\right)\left(\frac{K_{A_l}}{N}\right)\left(\frac{K_{A_c}}{N}\right)\right]$$

$A_q$ denotes the query event that is augmented by using word correlation mining algorithm. $A_l$ represents the location event, which only consists of a pair of latitude and longitude. $A_c$ denotes the calendar events composed of time and event description, which happen within a time window. That means there might be several calendar events for one submitted query. Likewise, the same query might be submitted in different place. Thus, several location events are possible to be associated with the query.

We traverse through all the query-result pairs. By repeatedly using the rule-mining algorithm, we can mine out the grammar rules for this particular user.

## Apply the Mined Rules

First, an arbitrary probability threshold is set, so queries will only be augmented to a certain level. Therefore, the length of the query can be bounded under a certain finite length. The threshold is set randomly because we do not know what is the best value.

The word correlation rules can be considered as a tree rooted at the first word, where each node can have multiple children. Therefore, finding out the most frequently co-occurring words of the query is the same as traversing the tree through the children nodes with the highest probability until the probability is lower than the threshold.

The most frequent correlated words always show personal interests. Therefore, we always pick the most frequently co-occurring word, so the query for this particular user will be personalized.

## Results

Based on the different situations as stated before, no conclusion can be drawn on which augmentation is better. Therefore, all the augmentations are implemented. When users submit a query, they will be given all the augmented queries with the raw query. Hence, users can select the most accurate query. After users finish searching, they will also be asked how good the augmentation was. Based on the statistics (the selected augmentation and the level of accuracy), we can learn which augmentation in general benefits user the most and how much it can benefit.

The statistics are shown in the following figure.

| Type: | Number | Percentage |
|---|---|---|
| Total Query | 291 | 100.00% |
| Raw Query | 264 | 90.72% |
| Successor Augmentation | 6 | 2.06% |
| Predecessor Augmentation | 15 | 5.15% |
| Highest Appear Repeats Augmentation | 6 | 2.06% |
| Location Augmentation | 0 | 0.00% |
| Non-Recursive Calendar Augmentation | 0 | 0.00% |
| Recursive Calendar Augmentation | 0 | 0.00% |

**Figure 6: Query Augmentation Percentage**

| Augmentation Type | Make No Sense | Worsen Query | No Difference | More Accurate | Best |
|---|---|---|---|---|---|
| Successor | | | 1 | 2 | |
| Predecessor | | | 4 | 8 | 3 |
| Highest Appear Repeats | | | | 3 | 1 |
| Location | | | | | |
| Non-Recursive Calendar | | | | | |
| Recursive Calendar | | | | | |

**Figure 7: Augmentation Evaluation**

Based on Figure 6, at the total 291 queries submitted, users use the raw queries at 90% of the time, which mean the augmentation does not have any effect on improving the query accuracy.

However, based on Figure 7 (the empty cell mean 0), once the augmented query is selected, the accuracy is very high. None of the augmentations has low evaluation. Moreover, the average of the augmentations makes the query more accurate. To make a more accurate conclusion, Table1 is constructed. Obviously, 60% of the augmentation makes the query more accurate. 20% makes the query the most desirable query for the user.

| | | |
|---|---|---|
| Total Evaluation | 22 | 100.00% |
| No Difference | 5 | 22.73% |
| More Accurate | 13 | 59.09% |
| Best | 4 | 18.18% |

**Table1: Evaluation Percentage**

## Conclusion

This application shows the potential of augmenting user's query. The reason that 90% of the queries user inputs are not able to be augmented is because the query randomness, or query ambiguousness. Consider the following sentence: John is an old friend of mine. It is impossible to conclude that John is old, based on the sentence only. The level of ambiguousness is even magnified when the query is processed by a computer. The reason is that human can use context to clarify what the sentence mean. However, it is extremely hard to train a computer to interpret the sentence in a certain context.

However, the augmentation algorithm has a very good effect on making the query more precise. Co-occurring words can strongly show personalities.

## Reference:


[1] ITU sees 5 billion mobile subscriptions globally in 2010, http://www.itu.int/ITU-D/ict/newslog/ITU+Sees+5+Billion+Mobile+Subscriptions+Globally+In+2010.aspx

[2] Email and Social Networking Most Popular Mobile Internet Activities, http://www.itu.int/

[3] Larry Page, Your Android battery life should last a day, www.techradar.com/news/phone-and-communications/mobile-phones/larry-page-your-android-battery-life-should-last-a-day-690439

[4] G. D. Abowd, et al. Cyberguide: a mobile context-aware tour guide. Wireless Networks, 3(5):421–433, 1997.

[5] N. D. Lane, et al. Hapori: Context-based Local Search for Mobile Phones using Community Behavioral Modeling and Similarity, Ubicomp 2010.

[6] S. Kandula, et al. What's Going On? Learning Communication Rules In Edge Networks, Sigcomm 2009.

[7] J. Yi, F. Maghoul, and J. Pedersen, Deciphering mobile search patterns: a study of yahoo! mobile search queries. In WWW '08: Proceeding of the 17th international conference on World Wide Web, pages 257–266, New York, NY, 2008.

[8] M. Arias, et al. Context-Based Personalization for Mobile Web Search. Internet, VLDB 2008

[9] C. Doulkeridis, et al. Querying and Updating a Context-Aware Service Directory in Mobile Environment. In Web Intelligence '04: Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence, pages 561-565, Washington, DC, 2004. IEEE Computer Society.

[10] J. Teevan, S. T. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pages 449–456, New York, NY, 2005. ACM.

[11] B. Smyth, et al. Exploiting query repetition and regularity in an adaptive community-based web search engine. User Modeling and User-Adapted Interaction, 14(5):383–423, 2005.

[12] Google. http://www.google.com/support/websearch/bin/answer.py?answer=134479