Scientific Journal Impact Factor (SJIF): 1.711

e-ISSN: 2349-9745
p-ISSN: 2393-8161

International Journal of Modern Trends in Engineering and Research
www.ijmter.com

INFLUENCING THE CONTEXT AWARENESS BY DESIGNING MOBILE SEARCH ENGINE

S. Anbalagan[1], T. Pravhien[2], A.Vinothkumar[3], Mr. T. Karthikeyan[4]

[1, 2,3]3rd Year, Department of Computer Science and Engineering,
Knowledge Institute of Technology
[4]Asst. Professor, CSE, Department of Computer Science and Engineering,
Knowledge Institute of Technology

Abstract--The rapid advance of wireless and portable computing technology has brought a lot of research interests and momentum to the area of mobile computing. One of the research focuses is on mobile search engines to leverage context awareness. With wireless connections, users can access information at any place at any time. However, the constraints such as limited client capability, limited bandwidth, weak connectivity, and client mobility impose many challenging technical issues. To keep up with this expansion, search engines the primary gateway to the Internet for more than half of all users must adapt to mobile environments.

In particular, search engines must take into account handsets, which are pervasive and person-centric, continuously capturing user-related information. Such data can provide more meaningful search results by augmenting searches with real-world information related to users' profiles and behavioral patterns. Information about users such as location, how they interact with the mobile device, or what's occurring in the surrounding physical world is called contextual data. Such information is increasingly accessible via sensors embedded in smart phones leading to many new commercial usage scenarios, such as location-based services and academic research on the topic. Our proposed model captures heterogeneous context data from many mobile sensors. We developed an application architecture that supports context-aware mobile searches, and use real context data from the Reality Mining project.

Keywords-Context awareness, leverage, search engine, Contextual data.

I. INTRODUCTION

Mobile computing is human–computer interaction by which a computer is expected to be transported during normal usage. Mobile computing involves mobile communication, mobile hardware, and mobile software. Communication issues include ad-hoc and infrastructure networks as well as communication properties, protocols, data formats and concrete technologies. Hardware includes mobile devices or device components. Mobile software deals with the characteristics and requirements of mobile applications.

Our investigation focuses on an extended form of mobile computing in which users employ many different mobile, stationary and embedded computers over the course of the day. In this model computation does not occur at a single location in a single context, as in desktop computing, but rather spans a multitude of situations and locations covering the office, meeting room, home, airport, hotel, classroom, market, bus, etc. Users might access their computing resources from wireless portable machines and also through stationary devices and computers connected to local area networks. We call this collection of mobile and stationary computing devices that are communicating and cooperating on the user's behalf a mobile distributed computing system.

II. RELATED WORKS

A. Data acquisition

The module samples the different hardware sensors and applies signal processing techniques to produce a context vector of clean signals with well-defined ranges. Typical operations at this stage include signal smoothing, scaling, thresholding, and applying stochastic filters such as Kalman filters. Raw context sensor signals are the inputs, whereas outputs are the normalized and cleaned context vectors.

B. Context reasoning

We apply a context model to infer context states from the underlying signal vector. The states we use are application dependent and need not be mutually exclusive. We can use both the activity contexts a user is in as well as various independent device states such as “low battery” and location. Context vectors are the inputs in this stage, while outputs are high-level context states.

C. State updates

After gaining a detailed and robust understanding of the context, the application can update its internal state. Aside from context, it can also use explicit user inputs and other application-state variables. With search, we can combine a user’s query with the context states to produce a contextualized search engine API query. In this stage, inputs are con-text states and user inputs, while output is the application state.

D. Contextualized output

Based on its updated state, an application can either produce contextualized outputs that are user-facing, as in the contextualized presentation of search results, or call external services in a contextualized fashion, as with augmented search engine API calls. The application state is the input in this stage, while outputs include user-directed interface actions and communication with other processes and applications.

III. PROBLEM STATEMENT

The computer user may move from one location to another, joining and leaving groups of people, and frequently interacting with computers while in changing social situations. One challenge of mobile distributed computing is to exploit the changing environment with a new class of applications that are aware of the context in which they are run. Such context-aware software adapts according to the location of use, the collection of nearby people, hosts, and accessible devices, as well as to changes to such things over time. A system with these capabilities can examine the computing environment and react to changes to the environment.

Pervasive computing simplifies life by combining open standards-based applications with everyday activities. It removes the complexity of new technologies, enables us to be more efficient in our work and leaves us more leisure time and thus pervasive computing is fast becoming a part of everyday life.

IV. SYSTEM DESIGN

We implemented our contextualization module using real-world data from the Reality Mining Project.

Data Attainment

This process is used for the using the search engine process. This process provides the preprocessing and the perfect information retrieval. Data pre-processing is an important step in the data mining process. A preprocessor is a program that processes its input data to produce output that is used as input to another program. The output is said to be a preprocessed form of the input data, which is often used by some subsequent programs like compilers. Typical operations at this stage include signal smoothing, scaling, thresholding, and applying stochastic filters.

Context Module

This module is provides the process of communication, location identification and application development. One challenge we face is that sensor data is highly heterogeneous, which makes it hard to form robust context inferences. In this process different types of applications are developed in the system. Applications are based on the acquisition process. Thus the process of mobile search process created efficient and effective.

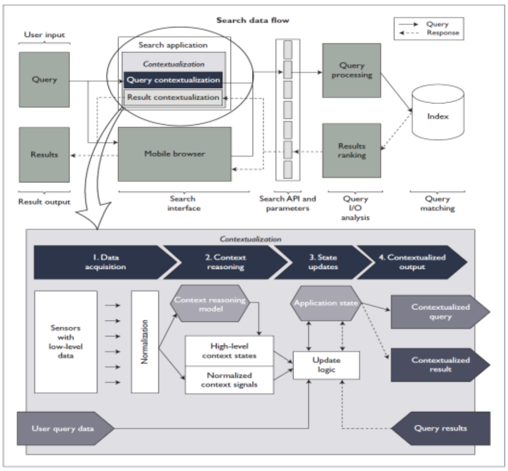

Fig 1 Mobile Search Flow

Here we propose a generic architecture that processes context signals and makes them available to downstream applications such as search engines through a contextualization module. This module comprises two components: query contextualization and result contextualization. a conceptual diagram of a mobile search application with our contextualization design. The contextualization module transforms the query text and context signals into the appropriate search API parameters. Once a user has queried the search engine via its API, the engine per-forms various internal query-processing steps such as spelling alterations, synonym expansion, and query classification — before sending the query to its index to retrieve relevant documents. The search engine then ranks the recalled documents by relevance to the query and returns them to the search application.

The contextualization module processes context signals in four stages: stages 1 and 2 are application-independent and solely concern acquiring context signals and modeling higher-level context states; stages 3 and 4 use application specific logic and I/O routines.

Context Model

The context model we describe next is based on the generic context architecture defined earlier, but uses only variables and knowledge from the RealityMining project. One challenge we face is that sensor data is highly heterogeneous, which makes it hard to form robust context inferences. We thus model the context signals as a unified context vector **V**(t), which is parameterized by time t . All context variables are synchronized with respect to time and bucketed into discrete time slices; the bucket duration determines how much context data is available for each context reasoning step. During real-time use, the bucket size also determines how quickly an application can respond to a change in context variables.

$$\mathbf{V}(t) = \{CS1(t), CS2(t), \ldots CSn(t)\}$$

where *CS* is the context signal value for time slice t.

We calculate some context variables as the number of times we observed the event associated with a signal in the time slice, which we measured as day of the week/time — that is, Monday through Sunday, morning, afternoon, or evening. Other context variables come from the application logs; we calculate these as the percentage of time the signal is on over the time slice. We use the context vector as the input to derive our higher-level context states. Our model design has two steps in which we define a set of rules. During step 1, we use the location cell ID to infer a location group, such as "Home" or "Work." In step 2, we use the derived location signal together with the remaining signals to infer the higher-level context states defined earlier.

Applying Context

Given the set of context states and the set of normalized context signals, we contextualize the Web search application (stages 3 and 4). Following our methodology, we distinguish between contextualizing the search API query and displaying the search results. As with our prior prototypical definition of context model, we can define an appropriate state update model for our search application in many ways. For this article's purposes, we mention several heuristic rules to illustrate the benefits that the available context signals have in different search scenarios. Although rules such as these are practical and effective in some cases, a more automatic and learned approach might be more suitable for other applications.

One important consideration when making context-sensitive augmentations to a search query is ensuring that the search engine isn't inadvertently altering the user's original intent. This is particularly important when we add additional query terms to a search. Because all the related logic must reside within the search application, the biggest challenge is how to make this decision without resorting to the internal knowledge the search engine maintains about queries, such as historical click-through information, intent classification results, and so on. One potential client-side solution is to use query similarity metrics based on the result sets alone. As described in research on query reformulation, we can define the similarity between two queries q and q' in terms of the similarity $s(r, r')$ of their respective result sets r and r'. In our case, we can compute $s(r, r')$ by counting the number of matching URLs as well as sub domains appearing within r and r'. Based on this idea, we can use the following scheme to determine whether a contextualization is appropriate for any given query:

- Generate contextualized and no contextualized query versions q and q'.
- Obtain search result sets r and r' for q and q', respectively.
- If $s(r, r') < t$ for an appropriate threshold t,

V. CONCLUSION

In this project we present a context-aware system that can be used with search engines to enhance queries. Although we focused on the common query-based mobile search scenario, our architecture is applicable to more general mobile computing scenarios. Our framework lets us combine heterogeneous sensor signals into a coherent context model. The next step will be a context-aware mobile search prototype; we're investigating machine learning approaches to replace the heuristic rules presented in this article.

On the context side, we're investigating using labeled context data to infer context states from the underlying signal vector. This will also help determine how much implicit or explicit user feedback is required to learn which queries and context states need certain contextualization in stage 3. User feedback will also help the system become more context-aware of individual user characteristics.

Using this kind of feedback-based learning will generalize our proposed model's applicability; heuristic rules will always remain application specific. This kind of model building would be greatly assisted by a standard language for marking up context states, whether to exchange annotated databases among researchers or data between applications.

Finally, our design has made the fundamental assumption that contextualization must happen largely outside search engines themselves, given that today's commercial search engines are largely unaware of important context signals (apart from location and time).

VI. FUTURE ENHANCEMENT

To support more powerful contextualization scenarios in the future, search engines themselves must become more context-aware. Besides user location and time, users' activities and other context information such as device characteristics will help search engines deliver more relevant and appropriate results for mobile user search queries.

REFERENCES

[1] J.L. Gómez-Barroso et al., Prospects of Mobile Search, tech. report EUR 24148 EN, Inst. for Prospective Technological Studies (IPTS), European Commission, 2010.
[2] J. Coutaz and J.L. Crowley, "Context Is Key," Comm. ACM, vol. 48, no. 3, 2005, pp. 49–53.
[3] H.W. Gellersen, A. Schmidt, and M. Beigl, "Multi-Sensor Context-Awareness in Mobile Devices and Smart Artifacts," Mobile Networks and Applications, vol. 7, no. 5, 2002, pp. 341–351.
[4] P. Korpipää and J. Mäntyjärvi, "An Ontology for Mobile Device Sensor-Based Context Awareness," Proc. 4th Int'l and Interdisciplinary Conf. Modeling and Using Context (CONTEXT 03), Springer, 2003, pp. 451–458.
[5] N. Eagle, A. Pentland, and D. Lazer, "Inferring Friendship Network Structure by Using Mobile Phone Data," Proc. Nat'l Academy of Sciences, vol. 106, no. 36, 2009, pp. 15274–15278.
[6] A.K. Dey, "Understanding and Using Context," Personal and Ubiquitous Computing, vol. 5, no. 1, Springer, 2001, pp. 4–7.
[7] L. Barkhuus and A. Dey, "Is Context-Aware Computing Taking Control Away from the User? Three Levels of Interactivity Examined," Proc. 5th Int'l Conf. Ubiquitous Computing (Ubicomp 03), vol. 2864, 2003, pp. 149–156.
[8] O. Lassila and D. Khushraj, "Contextualizing Applications via Semantic Middleware," Proc. 2nd Ann. Int'l Conf. Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous 05), IEEE Press, 2005, pp. 183–191.
[9] J. Yi, F. Maghoul, and J. Pede, "Deciphering Mobile Search Patterns: A Study of Yahoo! Mobile Search Queries," Proc. 17th Int'l World Wide Web Conf. — Refereed Track: Mobility (WWW 08), ACM Press, 2008, pp. 257–266.
[10] T. Heimonen and M. Käki, "Mobile Findex — Supporting Mobile Web Search," Proc. Int'l Conf. Human Computer Interaction with Mobile Devices and Services, ACM Press, 2007, pp. 397–404.
[11] K. Church and B. Smith, "Who, What, Where and When: A New Approach to Mobile Search," Proc. 13th Int'l Conf. Intelligent User Interfaces (IUI 08), ACM Press, 2008, pp. 309–312.

[12] H. Daume and E. Brill, "Web Search Intent Induction via Automatic Query Reformulation," Proc. Human Language Technology Conf. North Am. Chapter of the Assoc. for Computational Linguistics (NAACL-Short 04), 2004, pp. 49–52.
[13] K. Church and B. Smyth, "Understanding the Intent Behind Mobile Information Needs," Proc. 14th Int'l Conf. Intelligent User Interfaces (IUI 09), ACM Press, 2009, pp. 247–256.
[14] V.C. Storey, V. Sugumaran, and A. Burton-Jones, "The Role of User Profiles in Context-Aware Query Processing for the Semantic Web," Natural Language Processing and Information Systems, LNCS, Springer, 2004, pp. 45–62; doi:10.1007/978-3-540-27779-8_5.
[15] B. Begole, Ubiquitous Computing for Business: Find New Markets, Create Better Businesses, and Reach Customers Around the World 24-7-365, FT Press. 2011.
[16] M. Weiser and J.S. Brown, "The Coming Age of Calm Technology," Beyond Calculation: The Next Fifty Years of Computing, P.J. Denning and R.M. Metcalfe, eds., Coperni- cus, 1998, pp. 75-85.
[17] Y. Rogers, "Moving on from Weiser's Vision of Calm Com- puting: Engaging UbiComp Experiences," Proc. 8th Int'l Conf. Ubiquitous Computing (UbiComp06), LNCS 4206, Springer, 2006, pp. 404-421.
[18] B.J. Fogg, Persuasive Technology: Using Computers to Change What We Think and Do, Morgan Kaufmann, 2002.
[19] J. Müller et al., "Requirements and Design Space for Inter- active Public Displays," Proc. 18th Int'l Conf. Multimedia (MM 10), ACM, 2010, pp. 1285-1294.
[20] J.Müller et al., "Display Blindness: The Effect of Expec- tations on Attention towards Digital Signage," Proc. 7th Int'l Conf. Pervasive Computing (Pervasive 09), LNCS 5538, Springer, 2009, pp. 1-8.