UserIntent: Detection of User Intent for Triggering Smartphone Sensing Applications

Qiang Li*†, Qi Han§, Limin Sun *

* State key Laboratory of Information Security, Institute of Information Engineering, CAS, 100093

† University of Chinese Academy of Sciences, 101408

§Department of EECS, Colorado School of Mines, Golden, CO USA 80401

Email: {liqiang43,sunlimin}@iie.ac.cn, qhan@mines.edu

***Abstract*—User intent is an integral part of mobile phone applications as it delivers events to applications, notifies applications of relevant events, or triggers applications. Current smartphone applications either require users to manually start them or they run as background jobs. In this work, we propose *UserIntent*, a new paradigm for automatically selecting the right smartphone application based on user intent captured. UserIntent consist of two parts: user intent detection and mechanism for triggering a smartphone app. Action cues act as user intent and a context-aware selection algorithm chooses a suitable smartphone application. In order to demonstrate how UserIntent works, we also develop a concrete application that recognizes speaker and talk content based on gestures captured.**

***Index Terms*—Mobile sensing; Smartphone application; User Intent**

I. Introduction

Mobile phones are becoming a powerful platform for people-centric computing and playing a vital role in our daily life. Smartphone sensing applications are on the rise, in many of which people use their phones as a personal concierge service to capture various context, including people's behaviors [1], emotion [2], traffic conditions [3], talk conservations [4], and environmental noise [5]. In all these scenarios, however, smartphone applications either require people to manually start them when needed, or they run all the time in the background. We believe user intent is an integral part of mobile phone applications. User intent is a triggering message inside smartphones, notifying applications of various events including environmental changes, incoming data, and application events. In other words, a user intent is basically a message that you pass to applications, saying, "*Yo! I want to do...er...something!*".

To further understand user intent in smartphone applications, we first describe it via a hypothetical scenario. In the Poster and Demo session of a conference as shown in Figure 1, Alice comes to Bob's poster exhibition and listens to Bob's description of the poster. She finds the work interesting and wants to know more related or background work. She also would like to communicate with people who have worked on similar topic. Alice expresses her intent and expects a mobile application to provide her with timely and suitable information. The smartphone application is then turned on (and microphones, location sensors, or camera are also turned on), and the topic Alice is interested in is detected and similar content is recommended. In this example, user intent contains two most important pieces: the action and "something" to be done by smartphone applications. The Alice's "Like" is the action of user intent and Bob's talk is the "data" captured by a smartphone. Capturing and utilizing user intent is important for developing such novel applications.

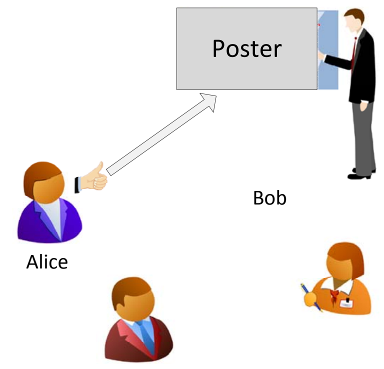

Fig. 1: The scenario of user intent at a Poster/Demo session. Alice indicates her 'like' intent of Bob's presentation, then a smartphone application is triggered to sense and recommend related content to Alice.

Research on understanding user intent is now emerging both in academia and industry. K. Church et al. [6] estimate search intent from user historical search behaviors, and J. Zhuang et al. [7] predict user intent on location based services (LBS) from the click-through history and location information. Google Now, EasilyDo and Xme are developed as personal Assistant Platform, exploring environment and receiving smart notifications about interesting things nearby to allow users to get instantaneous information about their surroundings. In this work, we focus on a fundamentally different usage of user intent. We aim to capture and utilize user intent, and deliver it to smartphone applications, thereby removing the need for manually turning on and off the applications or continually running the applications in the background.

II. UserIntent: A New Paradigm

The previous hypothetical poster/demo session scenario can be generalized to a broader problem space. We propose *UserIntent*, a new paradigm to automatically turn on the right

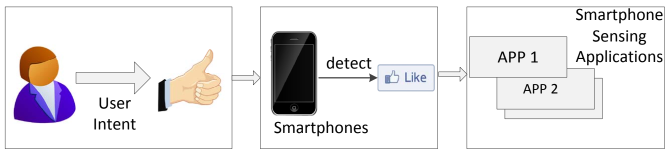

Fig. 2: The UserIntent paradigm

application using the captured user intent. Figure 2 depicts the general idea of UserIntent. A smartphone user first presents his 'like' intent for a certain thing. The phone then detects user intent and delivers it to smartphone applications. Finally one application is chosen and turned on for capturing user context.

To better illustrate the intuition behind the design of UserIntent, we analogize UserIntent to web services. In an $HTTP$ request, a pair $(verb, address)$ is explicitly declared, where the address (i.e., URL) indicates a resource (e.g., a Web page, graphic, or server-side program) and the verb indicates what should be done (e.g., GET to retrieve it, $POST$ to send form data to server for processing). Similarly, a user intent is expressed as: $UserIntent = Action + Object$, where the pair $(Action, Object)$ specifies the action that smartphone users present (e.g., gestures detected by phones) and the object specifies the smartphone application to be triggered. The difference, however, is that $UserIntent$ does not specify a particular smartphone application to access. Instead, a triggering mechanism is used to invoke an implicit call without explicit specification of applications.

This paradigm has two advantages:

1) It is unnecessary for users to remember the functionality of each application and when to turn them on and off.
2) Smartphone applications use embedded sensors, signal processing, and context model to recognize the context, incurring significant energy consumption on the phones. If these applications are triggered only when needed, there is a substantial energy saving compared with continuous running at the background.

While the idea of utilizing user intent to trigger smartphone application seems simple, many challenges arise in practice. First, it is not clear which application should be triggered. There are often conflicting or overlapping functions among smartphone applications, especially along with the increase of the number of applications. We cannot require a user to specify which application to start since that is no different from manually starting the application. Second, use intent detection itself is a smartphone application that continuously runs at the background. There is a latency between triggering and start of the application and some information may be lost during this time.

UserIntent contains two key issues: user intent detection and triggering mechanism of smartphone application. User intent can be expressed using mobile phones in different ways, including verbal cues (e.g., spoken words and sentences), action cues (e.g., gesture), and visual cues (e.g., eye contact).

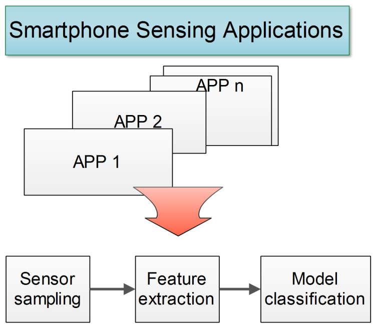

Fig. 3: Three stages of a smartphone sensing application.

The intuition behind the trigger mechanism is simple: functionalities of smartphone applications are directly related to the context recognized. For instance, CenceMe [1] detects people's behavior, EmotionSense [2] focuses people's emotion, and Nericell [3] considers traffic conditions. Context recognition usually includes three stages: embedded sensors sampling, features extraction, and models classification. As shown in Figure 3, we could infer functionality of applications based on sensors used by first dividing smartphone applications into different categories and then filtering.

Different applications use different sensors to sample raw data. For instance, CenceMe [1] uses accelerometer and Sociophone [4] uses microphone. It is straightforward to get sensor attributes that an application uses. We can continuously monitor embedded sensor reading when the phone is normally used, and then measure the sensor reading when the application is active.

However, sensor type itself is not sufficient to differentiate between applications. For instance, both Sociophone [4] and Earphone [5] use microphone, but they have different functionalities. To get features and model usage on phones in Figure 3, we apply a technical trick that measures energy consumption and computing overhead incurred by extraction of different features. The rational behind this design is that different feature extraction would consume different amount of energy and incur different computing overhead. As shown in Figure 4, we use microphone to sample raw data on Android platform (Nexus 4), extraction of different features consumes different amount of energy. For instance, Mel-frequency cepstral coefficients (MFCC) feature extraction consumes more energy than fast Fourier transform (FFT) feature extraction. We can infer feature usage based on energy cost and computing overhead. We also monitor changes in network traffic for different applications. Some complicated context sensing and inferring such as speech recognition or image processing would be offloaded to the cloud. The offloading leads to changes in traffic while the application is running.

The whole processing is as follows. When a new smartphone application is installed, the triggering mechanism invokes the application to run for a certain time duration and records the sensor type, features usage, and changes in network

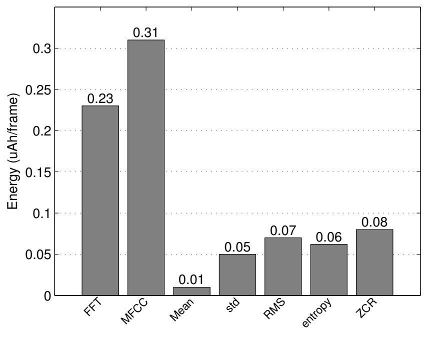

Fig. 4: Different energy consumption by extraction of different features from audio samples.

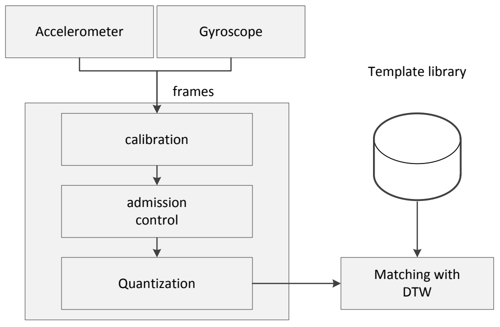

Fig. 5: Recognition of gestures as user intent cues.

conditions. This application is then closed and remembered as the specific functionality on context recognition. When a user's gesture is detected, the triggering mechanism invokes the application satisfying the needed functionality.

III. A Reference Implementation of UserIntent

In order to demonstrate how UserIntent works, we design an application that automatically turns on speaker recognition based on gestures captured. There are several reasons for choosing gestures as user intent expression. First, gestures are attractive for spontaneous interaction with people and mobile devices in the context of pervasive computing. Second, gesture recognition is immediate engagement and the overhead of setting up the recognition instrumentation is minimal. Gesture recognition couples data from accelerometer and gyroscope of smartphones and a gesture model. Unlike detecting verbal cues or vocabulary using speech recognition or detecting visual cues using computer vision, energy consumption and computing time is negligible for gesture recognition. Last, it is often desirable and necessary for users to create their own gestures or personalized gestures to represent their intent.

We next present key technical components of gesture recognition on smartphones. Figure 5 provides an overview of processing accelerometer and gyroscope data. We combine acceleration frames and relative speed frames to obtain smartphone movement trajectory by people's hand gesture. During preprocessing of sensor sampling stream, an admission control filters out some extraneous movements such as transition movements that are unimportant to gesture recognition. The rationale behind this is that intrinsic acceleration produced by hand movement does not change erratically; and rapid changes in acceleration are often caused by noise and minor hand shake/tilt. Admitted frames are passed to the Quantization stage, which reduces the length of input time series for Dynamic time warping (DTW) in order to improve computation efficiency. DTW is a classic algorithm based on dynamic programming to match two time series with temporal dynamics, given the function for calculating the distance between two samples. We employ the Euclidean distance for matching quantized time series of readings, and DTW calculates the matching cost and finds the corresponding optimal path. A template library stores one or more time series of known identities for each gesture, often input by the user. It recognizes the gesture as the best matched one in the template library.

Once user gesture is detected, a speaker recognition application is automatically started. This application is chosen as a result of filtering by microphone data sampling, MFCC and GMM.

IV. Conclusion and Future Work

In this paper, we propose a novel paradigm that uses user intent to trigger smartphone applications. We use gesture - a specific example of user intent cues, to illustrate this idea. In addition to user intent detection, we discuss how to design a triggering mechanism for selecting a suitable application. We will focus on two main issues in the future. First, we will extend the sensing application to general smartphone applications that run in the background. Second, we will design a middleware for managing when to open or close an application when needed.

References

[1] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM, 2008, pp. 337–350.

[2] K. K. Rachuri, M. Musolesi, C. Mascolo, P. J. Rentfrow, C. Longworth, and A. Aucinas, "Emotionsense: a mobile phones based adaptive platform for experimental social psychology research," in *Proceedings of the 12th ACM international conference on Ubiquitous computing*. ACM, 2010, pp. 281–290.

[3] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: using mobile smartphones for rich monitoring of road and traffic conditions," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM, 2008, pp. 357–358.

[4] Y. Lee, C. Min, C. Hwang, J. Lee, I. Hwang, Y. Ju, C. Yoo, M. Moon, U. Lee, and J. Song, "Sociophone: Everyday face-to-face interaction monitoring platform using multi-phone sensor fusion," 2013.

[5] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: an end-to-end participatory urban noise mapping system," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*. ACM, 2010, pp. 105–116.

[6] K. Church, B. Smyth, K. Bradley, and P. Cotter, "A large scale study of european mobile search behaviour," in *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*. ACM, 2008, pp. 13–22.

[7] J. Zhuang, T. Mei, S. C. Hoi, Y.-Q. Xu, and S. Li, "When recommendation meets mobile: contextual and personalized recommendation on the go," in *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 2011, pp. 153–162.