ISCHOLAR: A MOBILE RESEARCH SUPPORT SYSTEM

A Thesis

Submitted to the Faculty of Graduate Studies and Research

In Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Computer Science

University of Regina

By

Sojung Kim

Regina, Saskatchewan

December 2013

Abstract

The daily lives of researchers have been influenced in many ways by technological developments, especially mobile technology. Nevertheless, little attention has been paid to the formal adaptation of mobile technology into scientific research. *iScholar*, a mobile information retrieval support and recommender system for supporting research activities, is presented. The objective of this thesis is to integrate the latest, particularly mobile, technology when building effective research tools. Ubiquitously available connectivity and new features of mobile devices make it possible to create more flexible and portable research support systems. As a result, advanced research support systems will bring more efficiency and convenience when conducting scientific research. Web-based support systems are studied related to two dimensions - application and technology, and the same applies to mobile-based support systems. As such, the structure of this thesis consists of two parts - scientific research as an application domain and mobile technology as a technology dimension.

Three research frameworks that are taken from previous studies are reviewed to understand the application domain. One particular framework, the research cycle, is selected for further consideration. Possible mobile support for each step of the chosen framework is identified. The second part of the thesis addresses the technical details of how the system is designed and implemented. Literature review is considered as one of the most important research activities because it influences every aspect of the research process. Therefore, *iScholar* supports most steps of performing a literature review, such as searching and filtering resource databases, getting paper recommendations to read, and getting conferences/journals recommendations to target. The main functionalities of the proposed system are to retrieve publications as relevant as possible, recommend related papers to read, and recommend conferences to attend or submit papers to. By providing a useful tool, it is hoped that the system saves researcher's time and energy in carrying out research.

Acknowledgments

I am deeply grateful for my supervisor Dr. JingTao Yao for his encouragement, guidance, and inspiration throughout the years during my study. Without his numerous efforts and understandings, I would never been able to finish this thesis. I also would like to express my gratitude to the advisory committee members, Dr. David Gerhard and Dr. Orland Hoeber, whose constructive suggestions have helped me reshape this thesis.

I appreciate the Faculty of Graduate Studies and Research and the department of Computer Science for the financial support by giving me chances to work as a teaching assistant.

The work discussed in this thesis has also been inspired by many valuable comments and suggestions from my lab mates, Yan Zhang, Nouman Azam, and Ying Zhou.

Last but not least, many thanks go to the most amazing couple, Charlie and Jean Ottosen, for editing my English and giving me a lot of support.

Post-Defense Acknowledgments

I would like to sincerely thank the external examiner, Dr. Craig M. Gelowitz, for his invaluable comments and suggestions improving this thesis. I am also thankful to the defense chair, Dianliang Deng, for presiding the defense.

Dedication

Mom, I am sorry for all the times I hurt you. I do love you more.

To my sister and brother, Soyoun and Jongyeop Kim, your support, encouragement, and constant love have sustained me throughout my life.

To all of my friends who never stop believing in me. Thank you for waiting me with your moral support and endless inspiration.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The evolution of mobile technology has improved and supported daily activities in various ways. Likewise, with all the capabilities mobile technology offers, it promises to change the way of conducting academic research as well. Mobile devices, applications, and services have provided new ways of accessing and collecting data for researchers [34]. Yet few studies have explored mobile technology for scientific research. This research has been inspired by the WSS (Web-based Support Systems) and RSS (Research Support Systems) [72, 75]. I suggest that it is viable to extend the existing Web-based support systems and apply the new technology in building research support systems. While mobile devices include tablets, cell phones, audio players, and any other devices that are connected wirelessly, this thesis focuses on smartphones that have Internet capability and run on an operating system.

Research is a broad concept that could be used in different contexts. In this thesis "research" refers to scientific research (also scholarly research and academic research), which is a systemic way of finding answers to specific questions [9, 20, 33]. Kerlinger [47] established scientific research as "systematic, controlled, empirical, and critical investigation of hypothetical propositions about the presumed relations among natural phenomena." Research is a highly intensive human activity [75]. It may be difficult to formulate research procedures formally and provide holistic computational support [46, 75]. Even so, there exist some lessons and experiences that could be learned by new technologies, and it is feasible to support those parts of the research process.

With the fastest growing number of articles being published every year, scientific research requires significant effort to be up-to-date. Not only that, finding appropriate publications is dependent upon personal expertise or occurs in a somewhat unsystematic way, i.e., manually opening all the retrieved search results or refining search terms until satisfied. One of the biggest challenges in this information age is the information overload problem [6, 42]. By applying intelligent methods and Web-based technologies, information retrieval support systems help people reduce the amount of work required to define their information needs and evaluate retrieved results [40].

Recommender systems perform an integral part of reducing the information overload problem as well. Recommender systems (RSs) are software tools and techniques suggesting items that might be of interest to users. [64]. RSs assist decision-making processes, such as which movies to watch or what items to buy among many choices [64]. In that way, the number of items to be reviewed by humans decreases significantly. In the context of research activities, the items could include recommending literature to read, related conferences to attend, or appropriate experts to contact [8]. The proposed system might help researchers by providing personalized recommendations based on one's research interests, as well as reading histories and reflections. The system recommends which research publications are more relevant to one's study or what kinds of conferences one could possibly submit to or attend.

In this thesis I cover the adaptation of mobile technology with the purpose of producing more flexibility and portability to support systems. According to a Google survey in 2013 [32], over half of the US population now owns a smartphone and two thirds of the owners access the Internet through smartphones on a daily basis. The expectation is that mobile devices will probably soon overtake the personal computers as the most common platform [53]. However, many scholarly sites still do not provide mobile versions of their services, even though some fee-based services offer them. The proposed system, *iScholar*, is a new mobile system designed for academic researchers who are accustomed to online journal searching systems, but looking for more portable systems. It extends the online systems' availability to the mobile environment. Special requirements and challenges of mobile research support systems are analyzed and ways to possibly satisfy those requirements are suggested.

1.1 Motivations and Considerations

Web-based support systems (WSS) are a new multidisciplinary research area that deals with multiple application domains on the basis of Web technology [72]. In the technology dimension, WSS make the best use of the Web as the common platform, medium and interface [72]. On the other hand, from the application dimension perspective, WSS are evolutions of existing support systems into Web systems [48]. For example, if the Web technology is utilized to support the medical application domain, the systems can then be called Web-Based Medical Support Systems (WMSS) [48].

Web-based research support systems (WRSS) are the systems that support and improve research capacity, quality and productivity with Web technology [72, 43, 75]. According to Yao's classification [72], which defined a general level of WSS research as the study of WSS operations and support facilities, WRSS can be understood as a combination of the scientific research domain (an operation) plus the Web technology (a support facility). The support facilities are studied in terms of functional possibilities that computer science could offer, whereas the study of operations brings deeper understanding of the WSS domains [48]. As the name suggests, the main technological concern of WRSS is the Web as a support facility. An extension on technology dimensions may provide better support to any applicable domain. Therefore, I decided to integrate mobile technology into the support systems in order to extend the support facility from the Web to mobile, intuitively naming it Mobile Research Support Systems (MRSS) [48]. For the complete comprehension of the MRSS, both the scientific research domain (application dimension) and mobile technology (technology dimension) will be investigated in this thesis.

Almost every step of research activities can be aided by new technologies [46, 75]. Below I divide current uses of WRSS into three categories. The main objectives of the systems listed below are to support and improve research productivity [43]. Such systems will support scientists in finding relevant information, choosing the right research directions, and delivering productive results.

Exploring/retrieval - Exploring/retrieval of previous studies is performed in every stage of research [60]. One finds research-worthy ideas from literature in the early stage of research. One might even search through literature during implementation and writing stages. Information explorations differ from retrieval in that explorations browse the Web/databases with less concrete direction while retrieval looks for information in a specific context [46]. Examples of this category are online libraries and information retrieval support systems [75].

Reading - Reading is unavoidable in doing research, since research is built on prior findings [12, 60]. It can range from highlighting important findings to making comparative article summaries [46]. Reading support systems assist in mapping and managing relevant materials. For instance, a reading support system will allow for making bookmarks, as well as adding notes and links from different articles [75].

Writing - The use of appropriate languages, formal structures, and precise citations are required for academic writing [46]. Word processors and bibliographic management systems support researchers' writing activity. Future enhancement of writing support systems would be a system that recommends references automatically based on written text [75].

Tang et al. [43] argued that "The design and the implementation of viable Research Support Systems (RSS) depend on a clear understanding the research activities and process.[*sic*]" The understanding of research steps gives deeper understanding of support systems [17, 75]. For these reasons, I will first investigate three different research frameworks taken from previous studies. An appropriate framework will be selected to determine what kinds of research activities could be assisted. The one that is most understandable and intuitive will be selected. Sub-steps of the framework will be reviewed in detail to get an idea of how mobile technology can be used for each sub-step.

Exploring/retrieval support may be considered one of the most crucial because carrying out research inevitably involves researchers in a significant amount of searching and reading literature [12, 60]. O'Leary [60] drew up a list of working with literature over the whole research process: "choosing your research topic, developing your question, arguing your rationale, informing your work with theory, designing method, and primary data source." At each stage, literature might establish or change the direction of research. It is hoped that the proposed system might be a readily accessible tool for researchers to find relevant information with less effort.

Since there is relatively little understanding of mobile research support systems [5, 13, 18], there are a number of challenging issues that need to be considered including:

1. What would be the user scenarios when researchers are actually doing their research? Electronic resources such as journals and newspapers have been overwhelmingly increasing. How researchers could find appropriate resources and how computer systems could possibly support their effective findings?

2. Since it is not realistic for a system to support every aspect of research activity, it is therefore necessary to identify the more important research activities and functional possibilities of support systems. Since research is a combination of human-intense activities, what activities could be automated and what activities could not?

3. The technology mainly concerned in this study is mobile technology. What are the characteristics of mobile devices when they are utilized in this domain? What are the considerations when designing mobile research support systems?

4. Since the filtering and recommending of information procedures are complicated due to multiple aspects of the domain, is the traditional way of doing tasks still applicable? Are there more appropriate filtering and recommending techniques for supporting scientific research?

5. What would be the main characteristic that researchers want to derive from support systems? Some might prefer high performance, for instance fast returned information. Others might want to gather information with multiple languages/geographic locations. If the recommendations are the ones that the users want, how could the systems adapt to user feedback?

6. Using social media for research is an important trend as professional networking tools (e.g., Linkedin[1]) and social-citation sharing tools (e.g., CiteULike[2]) become known. What does social media via mobile mean for researchers and how would such tools help researchers to manage information overload?

The above list is not exhaustive. This thesis, for the most part, will only address the first three issues and slightly discuss the fourth. This thesis particularly proposes a mobile research support system for exploring/retrieval support and explores the functional possibilities.

Existing mobile tools intended for literature searching - iResearch[3], Sciverse Scopus Alerts[4], and Scholar Droid[5] - are surveyed and their limitations are considered. For instance, iResearch only provides a limited range of publications, and Scholar Droid does not provide full mobile interfaces - selecting an item of returned search result takes users to enter the publishers' website that may not be a mobile site. How the proposed system will tackle such limitations is proposed as follows.

- To provide a wide range of search results: the search results will be from multiple databases and various sources.

- To introduce new features: the system suggests new features such as providing more personalized publication recommendations or recommending conferences/journals to submit/attend.

- To reduce the physical restrictions: users can access academic resources whenever and wherever they want as long as the application is connected to the Internet. Users also can read downloaded papers off-line.

- To enhance user experience: the system serves a dedicated user interface for mobile systems.

Chapter 2

Research Frameworks

To determine what kinds of research activities iScholar could assist, it may be useful to begin with defining the research and research processes. Research needs to be distinguished from merely information gathering or re-organization of data [9]. Integral parts of research are to "interpret" the collected data and gain potential new insights [12, 20, 33]. Leedy and Ormrod [50] defined formal research as "in which we intentionally set out to enhance our understanding of a phenomenon and expect to communicate what we discover to the larger scientific community." The new insights could be gained through the researcher's personal experience, but it is more typical that the insights are attained through literature such as academic books or journals, namely literature review [60].

In his book, Anderson [3] gave a general description that "All research involves certain common elements such as defining the questions, reviewing the literature, planning the methodology, collecting and analyzing data, and disseminating findings." Even though there are common elements in the research process, it is important to note that not every individual research project follows the same sequence. Research methods and steps vary depending on academic domains and disciplines with no absolute classifications [60]. To start off, three different types of research frameworks from previous studies are introduced for a concrete understanding of research and research processes. Furthermore, one framework for further investigation is selected in this chapter.

2.1 The Research Cycle

Leedy and Ormrod [50] established a general research process as a cyclic step, as seen in Figure 2.1. The cycle begins with an idea, which then is refined to a concrete research problem [46]. In the research cycle, a researcher plays the role of producing information. It requires a researcher to establish a question and create his/her own answer(s). Cycling back repeatedly is the nature of discovery of knowledge because it is inevitable that the questions and plans will be revised and reconsidered throughout the process [9, 12]. It is not possible for a researcher to know every phenomenon in the universe. Therefore enquiries and disproofs towards existing knowledge continue. When an enquiry arises, there is a statement(s) to be proved. Such statement(s) turns into a potential hypothesis to be verified or solved through research.

The research process forms a cycle and follows developmental steps:

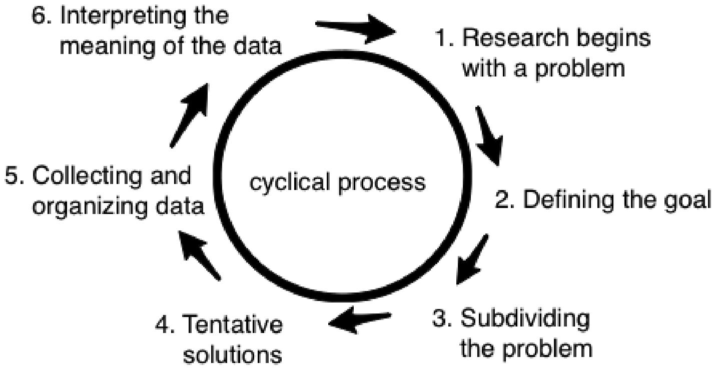

6. Interpreting the meaning of the data
1. Research begins with a problem
cyclical process
5. Collecting and organizing data
2. Defining the goal
4. Tentative solutions
3. Subdividing the problem

Figure 2.1: The research cycle (Source: [50])

1. Research begins with a problem: personal observations or experiences could form a tentative research idea at the first stage of research. The idea might be uncertain at the beginning, yet it has to be meaningful and realistic for one's study [20]. In general, it is desired to be a problem that an internalized topic based on one's accumulated knowledge.

2. Defining the goal: after considering the matter from various angles (feasibility study) a clearly stated research objective (an hypothesis or assumption) is defined in this step. The objective needs to be clear, detailed, and concise enough to explain the research and include key words that represent the study.

3. Subdividing the problem: viewing the problem through its subproblems helps researchers to get a clearer idea to approach the research. Each subproblem should not exceed the coverage of the main problem, as well as each covers significant areas of the main problem [50].

4. Tentative solutions: verifications for each sub-problem are conducted. Solutions to problems are tentative and to be judged by their usefulness.

5. Collecting and organizing data: experiments are conducted, and underpinning data for the tentative solutions is gathered and processed.

6. Interpreting the meaning of the data: the tentative hypothesis is either supported or rejected by the data.

Additional or new problems may emerge at the end of the cycle. One may start over from the beginning of the cycle or start a new research cycle with the newly established problem(s).

2.2 The Problem-Based Research

Ellis and Levy [21] placed a well-defined, research-worthy problem statement as the most important point on their *Problem-based research.* It views finding a worthy problem as the center of the framework and the starting point. They also stated that significant results are produced based on a well-articulated, well-supported, and well-argued problem. Figure 2.2 illustrates the problem-based research cycle. It also shows the interaction between the problem and the other elements of research.

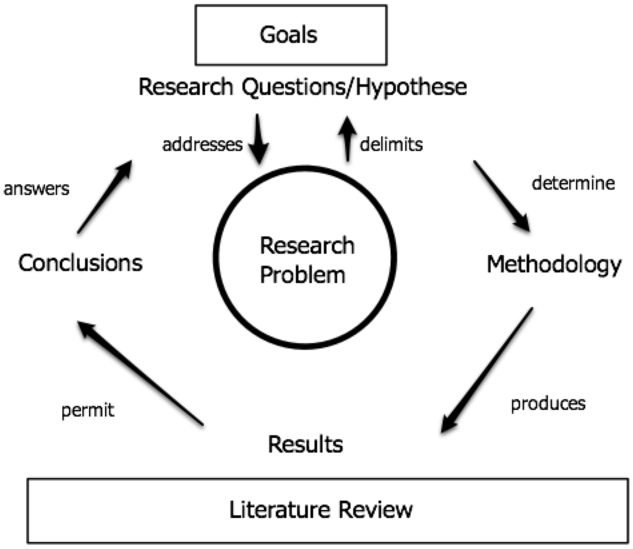


Figure 2.2: The problem-based research cycle (Source: [21])

Research starts with asking why the research is being conducted. Objects of scientific study are infinite and defining a subject is very important in many different fields. Only after a *problem* is proposed, can a subject be selected and a title of the study be established. By far the most important thing for researchers to do is to set up a question [50]. According to Kerlinger and Lee [47],"...adequate statement of the research problem is one of the most important parts of research." A noteworthy and meaningful problem leads to solid conclusions [20].

Research *goals* explain the intentions of research in addressing the problem. They answer the question “what does the study do?” Research develops theories by shaping rules to explain why or how a phenomenon happens. Also theories do not stay the same but keep progressing through small contributions from each study. The research eventually contributes to explain/operate/predict a phenomenon, connects an independent phenomenon to the whole structure, and expands the next study domain. Research *questions/hypotheses* are narrowed goals, moving towards answers to the problem. Therefore, the research problem addresses the research questions/hypotheses and, at the same time, delimits them.

Other elements are not directly related to the research problem, but the problem impacts them. Research questions/hypotheses determine the *methodology* that will take place to answer the problem. The methodology should give answers to when and where the research is conducted, as well as how and who does the research. The methodology produces research *results* such as data and interpreted information. Research results, in turn, permit one to derive *conclusions.* Eventually it could be said that the conclusions are the answers to research questions/hypotheses [21]. A viable research project should be based on the full understanding of an existing body of knowledge through *literature review.* No matter what a topic of research is, academic maturity is needed in order to internalize it. Academic maturity is not developed overnight but through long period of accumulated literature examinations and assessments.

As the name implies, the key element of this framework is *problem.* Ellis and Levy [21] detailed the four-step process of discovering research problems (Figure 2.3).

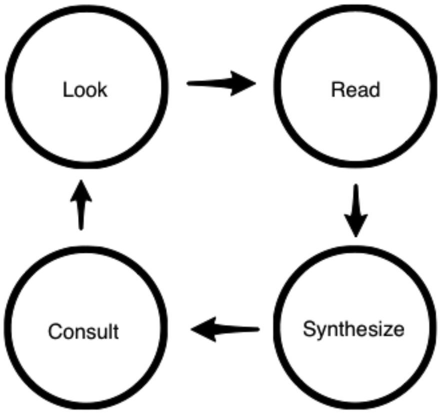


Figure 2.3: Finding a research problem (Source: [21])

1. Look: looking around and attempting to identify a potential problem, whether it is a personal experience or academic curiosity, is a starting point in conducting research.

2. Read: once one catches a *hunch* that it could become a good source of research, it is necessary to confirm by exploring existing knowledge. One builds knowledge and intellectual contexts in selected fields from literature review.

3. Synthesize: it was emphasized that only by the understanding of a number of works from different but related files is it possible to develop a worth-while problem.

4. Consult: after a tentative problem is established, experienced colleagues or supervisors can give feedback or do consulting. This step may either validate the problem, or return the researcher to the first stage.

The above two frameworks in Section 2.1 and 2.2 are designed mainly for social and behavioral science. Even so, I suggest that it is possible for them to be applied to other domains with few modifications. In the next section, one of domain-dependent approaches, especially in Information Systems (IS) field, will be reviewed.

2.3 Systems Development

Nunamaker and Chen [59] proposed systems development as both research domain and research methodology in the study of IS. Although there may be a controversy, the development of software systems is one form of research [59]. It is a way of research that involves creating something that previously did not exist [28]. Contributions to existing knowledge is achieved as a result of designing and building artifacts [39]. Ellis [22] pointed out that "...design and development research results in production of some form of artifact, and the process is indeed research, not to be confused with product development." "The pattern of this research progress is: 1) build a system, 2) develop theories and principles from observing behavior, 3) encode expertise in software tools for easy access, and 4) use these tools to help the development of new systems [59]." Figure 2.4 displays the systems development research process.

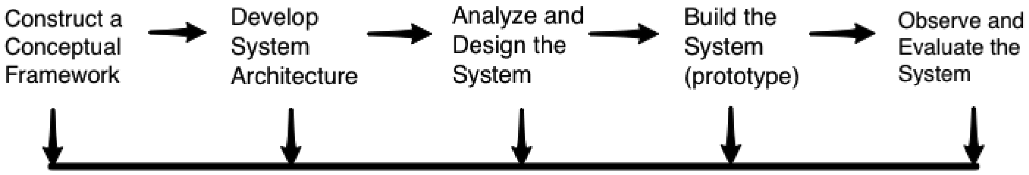

Construct a Conceptual Framework
Develop System Architecture
Analyze and Design the System
Build the System (prototype)
Observe and Evaluate the System

Figure 2.4: The Systems Development research process (Source: [59])

1. Construct a conceptual framework: a research question is developed if a tentative solution cannot be proven mathematically and tested empirically, or it employs a new method. A suitable conceptual framework is established in the first stage. Various relevant disciplines are also considered for fresh approaches and ideas.

2. Develop system architecture: system architecture arranges components, states system requirements, and specifies the relationships among the system components. It is enough to say that a good system architecture provides a good road

map for the system development process [59].

3. Analyze and design the system: system architecture is a road map of the system development process. Likewise, a system design gives a guideline for the implementation phase. The design of the database and modules for each functionality is defined at this stage.

4. Build the system (prototype): implementation of a system demonstrates the feasibility of the design and the usability of the proposed functionalities. The result of the implementation, whether it justifies or counterclaims the research, delivers new insights to the academic world.

5. Observe and evaluate the system: new theories might be discovered or advanced models could arise from observation and evaluation. Empirical laboratory tests, field experiments, or user surveys may be conducted for validation of the system.

The framework places a foundation in the current body of knowledge, namely knowledge of research methodologies and domains. Contributions to the body of knowledge are made by applying valid methodologies and understanding the domain. In particular, the framework has been adapted to iScholar system development. In Chapter 4, the framework will be introduced again.

Table 2.1 presents brief summary, scope and limitation of the three.

Table 2.1: Frameworks comparison

Leedy and Ormrod [50]	Summary	A research process is cyclical. Research originates with a question, requires well-planned goal, and is divided into manageable sub-problems. Data is collected and interpreted, and as a result tentative solutions are proved or rejected.
	Scope	The dynamic nature of research is highlighted in contrast to the conventional view that regards research as a one-time process. It is important to note that the end of one research cycle might open new questions or more research.
	Limitation	Research stops at some point for a variety of reasons - simply a subject is not interesting anymore, or some external factors such as funding or time restraints. Not every research continues to the end of the cycle, but many stops on the way.
Ellis and Levy [21]	Summary	Research problems are central to other elements of research. With literature review as groundwork, research problems, goals, methodologies, results and conclusions are closely interconnected and operational.
	Scope	The framework would be the general guideline for novice researchers, who are about to start their scholarly journey or who are easily lost during the journey.
	Limitation	Advised steps in the framework is not an exclusive way of producing a research problem. As more experiences are accumulated, a researcher tends to set his/her own approach in conducting research.
Nunamaker and Chen [59]	Summary	Systems development research begins with the conceptualizing of a problem. One's experience is accumulated during the design and development process. Eventually it produces artifacts for validation and evaluation.
	Scope	Systems development is the systematic research process used in the production of effective materials, systems, or artifacts. It was concluded that "systems development and empirical research methodologies were complementary to each other."
	Limitation	The application of the framework may be restricted to only domains which produce some form of output since not all research in various domains produces some forms of artifacts.

To sum up, there exists discipline-independent activities that WRSS can serve regardless of research domain, such as exploring, reading, and writing. Hence, Leedy and Ormrod's framework is selected for further scrutiny. Possible mobile assistants during the each step the research cycle is identified in the next section.

Chapter 3

Mobile Support for the Research Cycle

Advances in digital computing and communications are making a big difference to scientific research in multiple ways. Researchers are employing new technologies to setup a problem, solve the research question, and write/publish their findings. In particular, this thesis is interested in research support with mobile technology, for example mobile exploring/retrieval, reading, and writing support systems. The main advantages of using mobile devices for research are cost-efficiency and flexibility. Devices are becoming affordable, as well as most applications are distributed for free or an inexpensive price. Also, mobile applications are flexible and updatable to actively interact with users in real-time [62]. In Section 3.1 three major mobile characteristics that need to be considered when building a mobile application are discussed. Leedy and Ormrod's research cycle is re-introduced in Section 3.2 to analyze the possible mobile aids at each stage. Finally in Section 3.3, the importance of literature review, detailed steps of the literature review, and existing mobile tools for the literature review are discussed.

3.1 Mobile Characteristics

Before continuing discussion on mobile support for the research cycle, it would be worthwhile to point out the distinguishing aspects of mobile computing systems. Context-awareness, platform proliferation (large variety of platforms), and proactivity are the main characteristics that discriminate mobile devices from traditional stationary systems. It is crucial to understand the added dimensions of mobile systems and keep them in mind throughout the design and implementation process [10].

3.1.1 Context-Awareness

Context-aware systems gather context data and react to the current context without user intervention [56]. Context-awareness is one of the most beneficial factors that is driving the popularity of mobile devices [10, 54]. Mobile applications know where the user is located and current time of day/year. By setting up the places to match the user's preference, applications can deliver further refined services when it matters the most and optimize battery usage. Through sensing user interests, applications can match specific preferences and make better recommendations at the right time and place. Data transfer pricing, context data accuracy, and privacy issues are to be considered when planning to develop a context-aware system [19].

3.1.2 Platform Proliferation

Due to the high competition of producing mobile devices, proliferation of platforms that mobile applications have to support continues. Platform proliferation places a restriction on designing and implementing mobile applications. Supporting multiple platforms or considering platform-independence needs to be addressed when building a mobile application [10]. Another strategy would be not to build a monolithic application (i.e., include as many features as possible in one application), and to build small applications for each task with different platforms so that switching to a different platform does not become too complicated.

3.1.3 Proactivity

A proactive system detects events or situations and handles them without human supervision [67]. By merging contextual data such as location, time and preferences, mobile applications can deliver push communications to the right audience at the relevant time and place. When implementing a proactive system, considerations are: to make sure that users are able to protect privacy by the active interactions, to keep good user experience, and not to waste great deal of wireless bandwidth [10]. Mobile systems are challenged to observe users' continuously changing environment and send out alerts at appropriate time.

3.2 The Framework Decomposition

Leedy and Ormrod's research cycle is re-introduced in this section and analyzed to identify the possible mobile aids at each stage. Though the tools discussed are useful and applicable to any purpose, I find scholars in particular, could make good use of these applications while performing research.

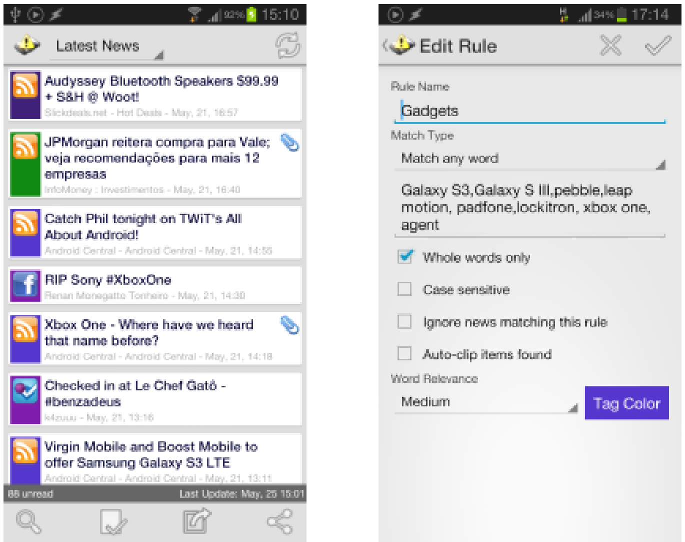
3.2.1 Beginning Research with a Problem Definition

The aim for a research problem is to contribute to a better understanding of a certain topic when it is answered [4]. There are many ways to develop research ideas and to recognize topics of interest to study, such as exploring literature, attending conferences, or seeking advice from experts [50]. Being up-to-date on various broad issues and keeping an eye on external phenomena are needed before choosing a specific topic [9]. In order to do these activities, a researcher might subscribe to online newsletters or explore literature in a variety of areas. For such activities, mobile technology could provide background study and brainstorming support.

Background study support

Using preferences to filter technical news could increase research efficiency. In another case, a researcher at a university could access library resources remotely from anywhere, saving a significant amount of time. Not only that, one can store interesting papers on mobile devices for reading anytime and anywhere.

Newspy (Android)[6] continuously searches several sources for relevant news based on custom rules, and notifies as soon as the articles arrive. Users define the desired news sources they wish to be periodically checked, such as RSS channels, Twitter timeline, webpages, or e-mail accounts. Then the application retains and alerts only those that match the rules defined. It helps tackle multiple information sources, i.e., the information overload problem, without missing any important news.



Audyssey Bluetooth Speakers $99.99 + S&H @ Woot!
Slickdeals.net - Hot Deals - May, 21, 16:57

JPMorgan reitera compra para Vale; veja recomendações para mais 12 empresas
InfoMoney : Investimentos - May, 21, 16:40

Catch Phil tonight on TWiT's All About Android!
Android Central - Android Central - May, 21, 14:55

RIP Sony #XboxOne
Renan Menegatto Tonheiro - May, 21, 14:30

Xbox One - Where have we heard that name before?
Android Central - Android Central - May, 21, 14:18

Checked in at Le Chef Gatô - #benzadeus
k4zuuu - May, 21, 13:16

Virgin Mobile and Boost Mobile to offer Samsung Galaxy S3 LTE
Android Central - Android Central - May, 21, 13:11

88 unread Last Update: May, 25 15:01

(a) News list

Rule Name
Gadgets

Match Type
Match any word

Galaxy S3,Galaxy S III,pebble,leap motion, padfone,lockitron, xbox one, agent

☑ Whole words only

□ Case sensitive

□ Ignore news matching this rule

□ Auto-clip items found

Word Relevance
Medium

(b) Edit rules

Figure 3.1: Newspy - Screenshots [6]

Brainstorming support

Brainstorming is an important process. It improves idea organization and provides a roadmap for a project. Combined with sketching, drawing, and graphic connections, it helps people make connections between previous and future knowledge. Drawing a mind-map or taking a short memo may help in organizing and capturing ideas, especially at the beginning of the research process. Mind-mapping is an organized approach to develop initial ideas [4]. The main idea is placed in the center and related topics are linked to the main topic [4] (Figure 3.2).

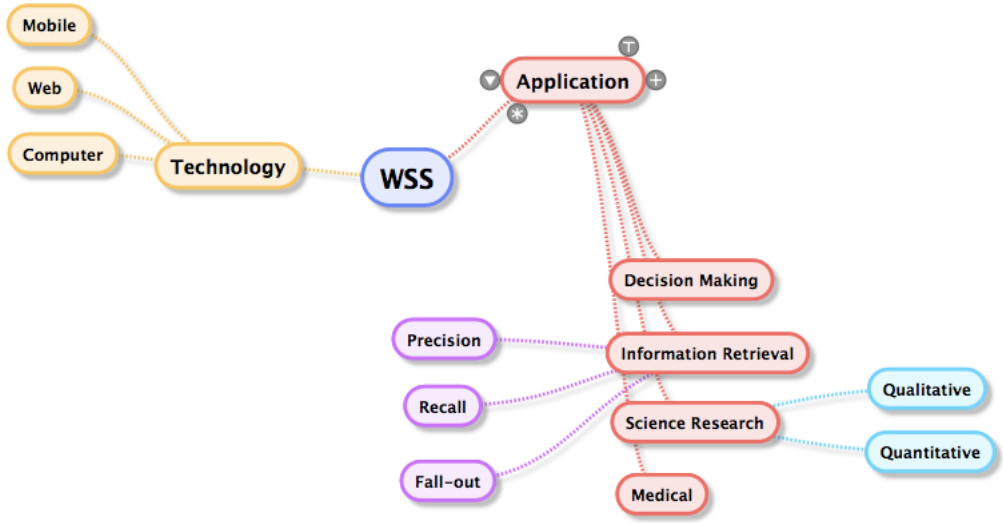


Figure 3.2: Mind-mapping

By visually displaying the connections between the main and other topics, mind-mapping helps in developing context for literature review [4]. With mobile mind-mapping systems, one can record ideas whenever and wherever they pop up.

Idea Sketch (iOS/Windows Phone)[7] is a mind-mapping application that provides diagram drawing, concept mapping, and flow charts drawing. It also makes it possible to convert the diagrams and flow charts to a text outline and vice versa. Lino (iOS/Android)[8] is an online sticky note board. It provides an easy way to take quick notes, reminder on special dates, and also share ideas with group members. The notes can include links, images, audios, and videos. With the applications it becomes easier to capture ideas anytime and share with colleagues.

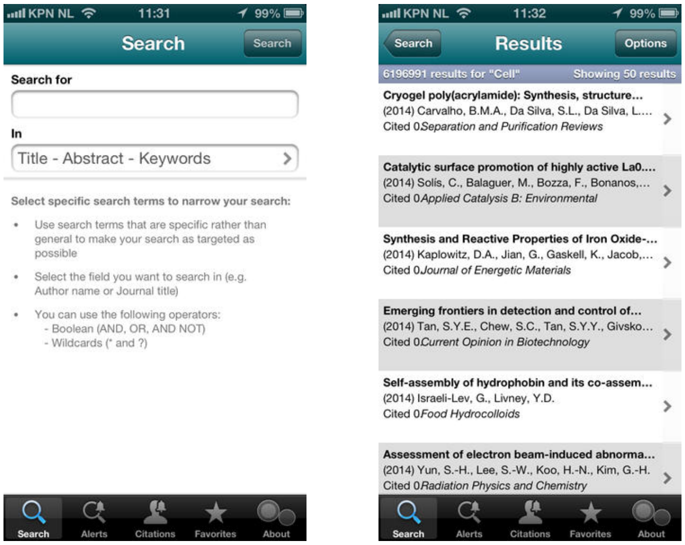
3.2.2 Defining the Goal

The main purpose of this step is to state a clear and complete goal of research. A good research question will be well focused, not too narrow, unambiguous, and realistic in the time frame [4]. Conducting a feasibility study and literature review, as well as writing a draft of research proposal is involved in this step. After considering the matter from various angles (feasibility study), a clearly stated research objective (an hypothesis or assumption) is defined. Browsing various relevant resources and planning schedules are the main activities at this stage.

Literature review support

Searching scientific papers, journal articles, and academic publications are among the key activities in literature review. New technologies may be taken advantage of for immediate and easy access to resources. Some universities and subscription-based databases, for example University of Regina library[9] and PubMed[10], provide mobile websites or applications for their students, faculties, or users. Complete listings of libraries and publishers offering databases with mobile sites can be found at the Library Success Wiki site[11]. A list of mobile applications for accessing scholarly information on mobile devices is examined in detail later in Section 3.2.3.

SciVerse Scopus Alerts (iOS/Android) [4] is an application developed for on the go access to the Scopus[12] database. Not only does it allow searches across Scopus articles and citations, but also the application lets users save the searches as alerts for regular updates of new research being published in their area of focus. Alerts can also be set up on the citation level, allowing users to see when a reference has been cited in other papers. It can also show other citation-level metrics for specific references.



Search for

In

Title - Abstract - Keywords

Select specific search terms to narrow your search:

- Use search terms that are specific rather than general to make your search as targeted as possible
- Select the field you want to search in (e.g. Author name or Journal title)
- You can use the following operators:
  - Boolean (AND, OR, AND NOT)
  - Wildcards (* and ?)

(a) Search

6196991 results for "Cell" Showing 50 results

Cryogel poly(acrylamide): Synthesis, structure...
(2014) Carvalho, B.M.A., Da Silva, S.L., Da Silva, L....
Cited 0*Separation and Purification Reviews*

Catalytic surface promotion of highly active La0....
(2014) Solís, C., Balaguer, M., Bozza, F., Bonanos,...
Cited 0*Applied Catalysis B: Environmental*

Synthesis and Reactive Properties of Iron Oxide-...
(2014) Kaplowitz, D.A., Jian, G., Gaskell, K., Jacob,...
Cited 0*Journal of Energetic Materials*

Emerging frontiers in detection and control of...
(2014) Tan, S.Y.E., Chew, S.C., Tan, S.Y.Y., Givsko...
Cited 0*Current Opinion in Biotechnology*

Self-assembly of hydrophobin and its co-assem...
(2014) Israeli-Lev, G., Livney, Y.D.
Cited 0*Food Hydrocolloids*

Assessment of electron beam-induced abnorma...
(2014) Yun, S.-H., Lee, S.-W., Koo, H.-N., Kim, G.-H.
Cited 0*Radiation Physics and Chemistry*

(b) Results

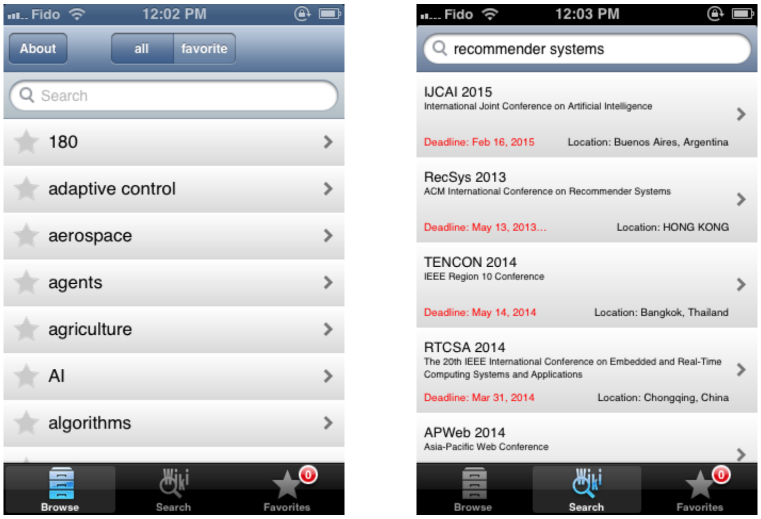
Figure 3.3: SciVerse Scopus Alerts - Screenshots [4]

Planning/managing support

For planning and managing purposes, mobile devices could be used for organizing tasks, assigning priorities and creating sub-tasks. Particularly for academic purposes, researchers could get upcoming conference alerts and manage schedules of conferences they are attending.

One can review or update a project's status from a mobile device when he/she is heading to a meeting. Existing management and collaboration tools provide a mobile version so the progress of a project can be checked wherever and whenever. Project Schedule (Android)[13] allows handling schedules right from mobile devices. With support for Gantt charts, the application makes it easy to allocate resources and keep track of cost and work for each task.

Call For Papers (iOS)[14] is an iPhone application that helps keep track of information on incoming conferences and journal publications. It allows users to search through numerous science topics and list corresponding calls for papers with mobile devices. It retrieves data from WikiCFP[15], which maintains the highest numbers of calls for papers in the science and technology fields.



IJCAI 2015
International Joint Conference on Artificial Intelligence
Deadline: Feb 16, 2015 Location: Buenos Aires, Argentina

RecSys 2013
ACM International Conference on Recommender Systems
Deadline: May 13, 2013... Location: HONG KONG

TENCON 2014
IEEE Region 10 Conference
Deadline: May 14, 2014 Location: Bangkok, Thailand

RTCSA 2014
The 20th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications
Deadline: Mar 31, 2014 Location: Chongqing, China

APWeb 2014
Asia-Pacific Web Conference

(a) Browse (b) Search

Figure 3.4: Call For Papers - Screenshots

3.2.3 Solving Sub-Problems

Since most research problems are too large or too complex to be solved at one time, a research problem is divided into several manageable sub-problems that make up the main problem. It is essential that each sub-problem should be consistent to the main problem and be expressed clearly and distinctly [50]. In doing so, sub-problems can be maintained in a controllable and progressive way.

Evolution/progress support

In order to make sure that all sub-problems are closely tied together, it is useful to make a graph of inter-related ideas or terminology. Furthermore, recording a history of changes for future structural modifications is also helpful. Since research is a rather cyclic and ongoing process, it would be easy to fall off the tracks as the study progresses [50]. New technology keeps research on track by supporting the storage of ongoing reflections and the integration of research findings.

Beakr (iOS)[16] is an experiment check-in application. It allows researchers to check on their experiments day to day and when completed. This application allows users to track the frequency of their experiments and its success/failure rates. The application has a built-in list of common ordinary tasks and experiments in science labs. Users can easily check into these experiments on the list and can also upload an optional photo for each check-in.

Sohn et al. [26] proposed a prototype of a location-based reminder system, named Place-Its. Upon arrival or departure to a place which is previously named as one's office or classroom, pre-set messages are sent to his/her mobile devices. Location-triggered reminders might be useful to organize and manage one's tasks while juggling multiple priorities.

Experiment/data collection support

Surveys and interviews could be distributed and received through mobile devices. It may become convenient to control experimental variables and regulate research simulations as well. The more smart phones are used, the easier for researchers to track experiments remotely and maintain contact with laboratory members.

When doing bench work, various applications can help to reduce time, such as OD-260 (iOS) [17], LabTimer (iOS)[18], and Notes Plus (iOS)[19]. Mobile devices work as a unit converter, calculator/timer, and notebooks in hand all the time [61]. Detailed lists of laboratory and scientific applications can be found at Scientific Mobile Apps[20] page and Mobile Science page[21] for chemistry researchers.

3.2.4 Interpreting the Data

The last activities of the research cycle would be interpreting collected data and actually writing the results of the research. Interpretations of data present grounds for supporting or rejecting the tentative hypothesis. The advance of the Web provided a new platform for communication between scholars, simplifying communication and collaboration [46]. Mobile applications offer instant communication and easy bibliography generation.

Communication support

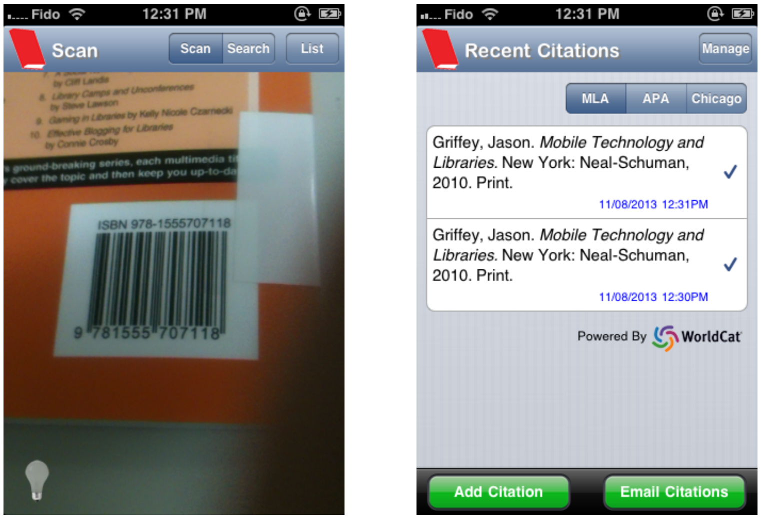
Mobile devices can support instant interaction for analyzing data and exchanging opinions. Affordable long-distance communication among individuals in the form of mobile video calls or conference calls could be used. In some cases, social-media can also be utilized for such purposes.

Adobe Connect Mobile (iOS/Android/Blackberry)[22] gives users a complete platform to host and participate in mobile conferencing. Presentations can be interactive for people viewing on a mobile device, letting them tap in questions and answers to give their input. Also, video conferencing lets the users view both their own video feed as well as the person talking, on top of viewing phased feeds of everyone else in the meeting.

Writing/publishing support

For writing and editing support, possible extensions would be mobile reference management systems for managing and writing bibliographies. In addition, mobile technology could support these activities by alerting the users of upcoming calls for papers and recommending relevant journals/conferences to submit to.

EasyBib (iOS/Android)[23] enables users to scan a book's barcode, record the information necessary, and create a proper citation for their writing. Users can add citations on the application or email those citations to themselves from the application.



Griffey, Jason. *Mobile Technology and Libraries.* New York: Neal-Schuman, 2010. Print. ✓
11/08/2013 12:31PM

Griffey, Jason. *Mobile Technology and Libraries.* New York: Neal-Schuman, 2010. Print. ✓
11/08/2013 12:30PM

(a) Scan a barcode

(b) Create a citation

Figure 3.5: EasyBib - Screenshots

Dragon Dictation (Android)[24] is useful for recording ideas or sending a reminder to oneself without having to type on a keyboard [35]. Dictation might be a solution for the major issues of mobile writing support [5], although accuracy and efficiency over traditional ways are not yet proved.

Table 3.1 lists the introduced applications and compares whether the considerations of mobile characteristics in Section 3.1 match.

Table 3.1: Mobile research support tools matrix

	context-aware	variety of platforms	proactive
Newspy[6]			○
Idea Sketch[7]		○	
Lino[8]		○	○
Sciverse Scopus Alerts[4]		○	○
Project Schedule[13]			○[*]
Call For Papers[14]	○[**]		
Place-Its [26]	○		○
EasyBib[23]		○	
Dragon Dictation[24]			

3.3 Review of Literature Review

As recognized in the previous section, research activities cover a large spectrum [43]. It is rarely possible for one application to assist in every aspect of research. Therefore, it is determined to design and develop iScholar to assist one of the most crucial activities, that is literature review. Literature review is very important in research steps [60], because, according to Boote and Beile [15], "a thorough, sophisticated literature review is the foundation and inspiration for substantial, useful research." Literature is used to familiarize the researcher with a new study area, monitor the study area to avoid duplication, and develop reference lists [23]. Not only that, for a practical purpose, literature review is required for proposals for funding or a degree [29]. Since literature review is mostly related to the exploring/retrieval category, that category is selected as the target subject of iScholar. The importance of literature review is briefly reviewed in this section and also potential mobile technological support for the literature review are identified.

[*] Reminder

[**] Related CFPs

3.3.1 The Importance of Literature Review

A literature review is a systematic method for identifying, evaluating, and synthesizing previous studies by other researchers [29, 4]. The ultimate purpose of literature review is to integrate where an author agrees or disagrees on what is known in an area [15]. One builds knowledge and intellectual contexts in selected fields through it [51]. In other words, literature review tells how research has been carried out up to now and what is missing where one's research intends to fill in [51]. Another aim of the literature review is to clarify one's ideas and findings, and, by doing so, it justifies the importance for a new study [51]. Even though one's study is built upon the current literature, that does not mean that literature review is merely summarizing or just explaining what other researchers have done [50]. Rather, it gives rationales for why research study needs to be performed based on previous works.

Literature could influence every step of research [60]. At each stage of the research cycle, literature might establish or change the direction of research. On some occasions, literature also frames valid research methodology, approach, and goals [51]. In spite of its importance, conducting a good quality literature review is not a simple task, especially in emerging fields because the nature of such fields is diverse and interdisciplinary. For instance, the field of machine learning is basically rooted in computer science but also deeply concerns artificial intelligence, robotics, natural language, and others [1]. Rather than relying only on computer science indexes, searching other indexes that cover cognitive or social sciences may bring out other aspects in the field [1]. In that case, literature should come from a variety of sources, in other words multi databases and various conference proceedings [51]. It is advised for researchers to utilize multiple databases in order to avoid narrow and shallow background depth [51]. It will be discussed in a later chapter how iScholar assists in building related knowledge and also how to deploy multidisciplinary databases for searching literature.

3.3.2 Exploring/Retrieval Support for Literature Review

Literature review is an essential tool for synthesizing previous research and information on a topic. It has to be undertaken in a systematic manner - identifying all the available literature on a topic comprehensively - so that the conclusions of the review are dependable [4]. Once a problem arises, one develops a exhaustive searching strategy, sets inclusion and exclusion criteria to ensure that only relevant papers are included, and critiques the selected papers based on the criteria [4].

Undertaking a systematic literature review requires much effort and is time-consuming. It is important to conduct literature review with a comprehensive search strategy to give confidence to readers that all relevant literature was reviewed [4]. Fortunately literature searching has been revolutionized by the evolutions in electronic databases. One could get advanced technological aids for each sub-step of literature review with mobile devices. The list below explains the detailed sub-steps of literature review with examples of mobile support. Exploring/Retrieval support for the sub-steps with mobile devices involved in literature review are highlighted.

1. Performing a search: numerous studies are printed in journals, books, magazines, and the Web each year. This stage is to explore literature and identify valuable studies. Online literature review sources are public/private/specialized bibliographic databases, university libraries, and Web search engines. According a study [13], 10.4 percent of survey respondents (university students) most frequently use smartphones for academic purpose such as applications, search engines, online encyclopedias, and libraries. Seventy-six percent of them answered that they use applications mostly for finding academic information.

The study shows that the use of mobile applications for information seeking is increasing, especially among young scholars and academics [13].

2. Organizing collected literature: researchers deal with thousands of resources. Therefore, it is important to stay organized and keep files in order. Online reference organizers help researchers store their searches and references in a virtual file cabinet [29]. Using these tools may require less effort for maintaining collected literature in an organized way. It is possible to synchronize personal computers with mobile systems, so that organization can be maintained the same in every device used by an individual.

3. Reading and screening collected literature: to identify the most relevant studies that are related to a researches' own research topic, one has to read throughly the collected papers and gradually decrease the size of the collection. Most online scholarly literature is in portable documents format (PDF), which can be opened and viewed exactly the same on any system. Since most mobile OS (Operating Systems) already support reading PDF files, researchers can download files to the systems and read the files at any time.

4. Taking notes on important findings: one objective of reading literature is to obtain background knowledge on a selected topic. It is worthwhile to develop a systematic approach to review literature and record summaries/findings in an organized way [60]. Some might prefer using different colors for different categories or others might keep a research journal. In addition to the portability and synchronizing with cloud services, mobile systems provide simulating handwriting, tags (makes searching through notes simpler), and audio recording [25].

5. Listing and maintaining references: once most relevant studies are screened,

reference lists for future writing needs to be maintained. There are many popular citation management tools, such as RefWorks[25] and EndNote[26]. These tools could be extended to mobile systems to give more freedom to physical limitations. Backup is very important for not only writing up but also managing collected references. Cloud services like Dropbox[27] let users sync folders on their personal computer(s) with mobile devices and save previous versions.

6. Writing a literature review: once a literature review is completed, the findings need to be presented, both in oral and print if possible. Though dictation tools might help make quick notes or emails, how mobile systems will support lengthy writing is uncertain yet because of input challenges [5]. iScholar supports this task in a realistic way by recommending conferences and journals to which writing can be submitted.

3.3.3 Mobile Tools for Literature Review

Modern mobile technology might be useful for academics and researchers by enhancing their work efficiency. Particularly one of them would be the speed of the search process. Knowing the right tools for searching is half the success for finding quality information [1]. The other half is how to use such tools effectively [1]. This section introduces existing useful mobile applications for literature searching - iResearch, Sciverse Scopus Alerts, Scholar Droid -, as well as the limitations of each application.

iResearch

iResearch (iOS)[3] connects mobile devices with full text access of American Institute of Physics Journals. Once PDFs are saved locally users can read the article off-line. It requires an individual or an institutional subscription to access full text. The main objective of this application is to provide full texts of AIP articles. Hence, contents are limited to selective journals. Currently, this application includes 13 journals. Since the application does not provide a search feature, one has to know the exact volume and issue numbers and find an article manually by the numbers. Furthermore, to get full access to the service, one must have either an individual subscription or institutional subscription (via IP authentication). Otherwise, one has to pay per one full-text view.

Sciverse Scopus Alerts

Sciverse Scopus Alerts (iOS/Andoroid)[4] searches the Scopus database for article abstracts and citations. Users can get search and citation alerts via the application. It also allows users to write notes and to append them to an article for use back in the lab and share articles with colleagues with questions/comments attached. Limitations are full-text is not available through this application (only links to citations) and one cannot see more than 50 search results for any search. Also the service is available only to existing subscribers. To get full access, one must be registered with an institution email and be affiliated with a subscribing institution.

Scholar Droid

Scholar Droid (Android)[5] is a free application that lets users conveniently browse through search results returned by Google Scholar and look up book results on Amazon. It provides a useful interface for retrieving results from Google Scholar, which provides comprehensive academic resources. But the results of the search include only titles and summaries of papers. Clicking on an article title enables users to enter the publishers' website that may or may not be mobile friendly.

Table 3.2 lists what tasks of literature review are supported by introduced applications and what are not. iScholar tries to fill the limitations of the applications so may be used for every steps of the literature review activity.

Table 3.2: Mobile literature review support tools

	iResearch[3]	Scopus Alerts[4]	Scholar Droid[5]
Performing a search		○	○
Organizing collected literature	○		
Reading and screening collected literature	○		○
Taking notes on important findings		○	
Listing and maintaining references		○	
Writing a literature review			

Below I propose how iScholar will tackle such limitations.

1. Range of data: by using Google Scholar as a database, the system will be able to provide a wide range of search results, unlike iResearch which only gives selective contents.

2. Readability: by using dedicated user interface the system will enhance readability. Scholar Droid provides an interface for Google Scholar but only for keywords searching with summaries display.

3. New features: iScholar includes some more unique features such as recommending conferences/journals to submit/attend, downloading/bookmarking interested publications, and adding comments on downloaded files.

Besides providing multiple database searching and mobile dedicated interface, I also propose additional features such as recommending papers to read, and recommending conferences/journals to submit/attend. With the rapid increasing journals and conferences each year, researchers now face the information overload problem with selecting papers to read and conferences to submit. I suggest that such features will enhance literature review productivity in general, particularly in performing a search and writing a literature review.

In conclusion, the significance of literature review was identified in this section. Possible mobile technological aids for literature review were recognized as well. I have implemented a mobile application for this purpose. In Chapter 4 and Chapter 5, the design of the system and prototype implementation are discussed.

Chapter 4

Mobile Research Support Systems Design

This chapter discusses the iScholar system design. Building a system follows a specific process such as developing system architecture, analysing requirements, and evaluating the system. Accordingly, the analysis of system functionalities and requirements, architecture of mobile research support systems, and detailed system design of iScholar are discussed in order.

4.1 Systems Development Framework Revisit

It is the purpose of this thesis to demonstrate that it is practical to build useful mobile research support systems. Accordingly, I employ the *Systems Development* framework as the methodology of this study and so bring attention back to the framework. The general steps are to define the functionalities, requirements, architecture, and design of the system in order. The below process is adapted from Nunamaker and Chen's article [59]:

Phase 1: Construct a conceptual framework

- Identify the system functionalities - Section 4.2

- Investigate the system requirements - Section 4.3

Phase 2: Develop system architecture

- Develop an architecture design - Section 4.4.1/4.4.2
- Define system components and inter-relationships - Section 4.4.3

Phase 3: Analyze and design the system

- Design the processes to carry out system functions - Section 4.5.1
- Design the database - Section 4.5.3
- Develop alternative solutions and choose one solution - Appendix A.1

Phase 4: Build the system (prototype)

- Learn through the system building process - Section 5.1/5.2
- Gain insights - Section 5

Phase 5: Observe and evaluate the system

- Evaluate the system - Section 5.3
- Consolidate experiences learned - Conclusion

4.2 System Analysis

Historically, with the advent of the Web literature searching moved from paper searching to online searching. I believe we are now in the second dramatic change of literature searching from the Web to mobile. Usage and familiarity of mobile devices are growing very fast, so in the near future mobile may become the only personal computing platform. That is the chief reason I propose it is time to shift support systems from the Web to mobile. iScholar is an attempt to accept the change.

The primary purpose of iScholar is to support literature review activities, mainly providing exploring/retrieval support. Yao [74] suggested that in the context of libraries there have been attempts to automate the task of searching information by, for instance, providing keyword search or visualized search results. Even so, support functionalities such as filtering search results or recommending proper resources are still left to librarians. iScholar focuses on supporting analyzing, organizing, and discovering tasks which previously were the tasks of librarians, rather than simply responding to users' queries and displaying the retrieved data.

Current Web browsers are used as exploring support tools [43]. As a part of efforts to address individuals' specific needs based on the Web, there are research article indexing systems such as Current Content[28] and DBLP[29], and citation analysis systems such as SCI and CiteSeer[30] [73]. I try to extend the availability of these Web-based support systems to the mobile area. Although there have been some attempts as introduced in the previous section, most existing applications remain as supplementary tools. I expect that the main reasons are limited coverage, limited access to data sources, and insufficient supporting functionalities compared to Web-based systems. iScholar, the proposed system, will provide multi-databases searching/retrieving academic data and personalizing of the retrieved data.

Given a mobile device already holds information about individuals (reading histories) and if a user actively adds some additional data (academic interests), then the system will be able to: 1) recommend a conference or journal to attend related on personal interests plus upcoming event dates and 2) recommend a conference or journal to submit based on written text which one has been working on. As for additional helper features, I also propose two more functionalities: 1) the system will provide ways to bookmark or download selected papers into mobile devices, make notes on important content, and connect articles with each other 2) by utilizing the personalized information of individuals such as academic interests and reading histories, the system will recommend related articles to read in the future.

Following is the summary of proposed features of the system in accordance with above analysis:

1. Multi-databases searching for discovering and retrieving academic information.
2. Adjusting the order of the retrieved data based on personal information.
3. Recommending a conference or journal to attend related on personal interests plus upcoming event dates.
4. Recommending a conference or journal to submit based on written text that one has been working on.
5. Downloading selected papers into mobile devices, this will be especially useful if one downloads papers when he/she has a direct access to full contents, e.g. within a school or library, and wants to read the contents anywhere.
6. Bookmarking papers, make notes on important contents, and connect articles with each other.
7. Recommending related articles to read in the near future based on one's academic interests and reading histories.

4.2.1 Context in Mobile Research Support Systems

As users become familiar with mobile devices, the users want to use their devices for any tasks on the move. This fast shift to mobile environment poses new challenges for research support systems where context is a stronger trigger for information needs and where the context changes rapidly. In this section how context is defined in a wider sense and in mobile research support area is introduced. The nature of mobile search with the context and possible approaches of mobile research support are followed in the next section.

Context data provides a good way of discovering and understanding a user's information needs [31]. The definition of context varies from domain to domain. Each discipline defines context in its own way or point of view [31, 64]. Myrhaug and Göker [58] proposed a comprehensive framework for exploiting context. Context related a user consists of five sub-contexts: environment (entities surrounding a user), personal (physiological and mental), task (what the user is doing), social (friends or family, and their presence), and spatio-temporal(time and location) context.

From the perspective of mobile computing, context was initially defined as a user's physical location, human neighbors around the user, object neighbors, and the relocation of these elements [64]. Other elements, such as date, season, temperature, physical and conceptual statuses of interest, emotion and so forth, have been added to the definition of context in terms of mobile computing.

Any data that is gathered by devices could be defined as contextual data, such as locational or environmental data. Some contextual information can be directly taken from raw data, while others need to be inferred from a user's profile. Winoto et al. [70] proposed a paper recommendation system that considers context when recommending papers to graduate students. They said, in the field of e-learninag, context includes users' learning goals, background knowledge, motivations, and so on.

Possible use of context while recommending research articles and conferences using mobile devices is proposed below.

- Previous publications/Academic interests: a researcher's previous publications and academic interests give a good hint for a support system to tell specific research topics that the researcher wants to work on and how the researcher's academic interests have been changed (based on the publication history).

- Downloaded papers/Reading history: whereas the first two properties may be rather static, these two properties shows the latest interests of the researcher. Also, since researchers tend to read a variety of papers from various sources, these two properties may bring a wider variety of contextual information than the previous ones.

Context-awareness involves complexity such as monitoring contextual data, uncertainty modeling, real-time processing and merging multiple and, probably, disagreeing sensor data. A major challenge is that the information interpreted by such context-aware systems must be making sense to the user, or at least there needs to be an explanation the user can access.

4.2.2 Mobile Search

Mobile search is different from the traditional search in two ways. One is that the search is performed in a dynamic environment, and the other is that mobile search tries to make information relevant to the environment (i.e., contextual) [31]. Location-based search, which tailors retrieved results by where a query was sent, is widely used at present, for instance, local news, nearest points of interests, and weather services. Time and preferences are another good sources of mobile search [31]. Retrieved information may be filtered by time and preferences to provide more satisfying results. Further, filtered information based on the locations, time, and preferences can be pushed to users (i.e., proactive). Mobile systems can keep track of users' situation around, and deliver push notifications when they decide that the situation is important enough to alert.

The main consideration is to identify correlations between certain application domain and context data. For example, how contextual data (e.g., location, time, or preferences) can be used in the context of research support systems needs to be defined. The below are some examples of how a research support system could use the context-awareness and proactivity from mobile devices.

- If a searching occurs in Korea, he/she might be more interested in publications that are written in Korean (geographical). Or if one keeps searching papers written in Korean, the system could display papers written in Korean in the higher positions (linguistical).

- A research support system might be able to recommend conferences that are held around a user's location (locational context). Or a conference recommender system could keep the detailed schedules of conferences and alert when the preferred speakers give a talk (temporal context).

- A system can notify users when the system suspects some other papers that have plagiarized the owner's paper (by preferences), or when the deadline for submitting to conferences that the owner might be interested is coming up (by time).

4.3 System Requirements

iScholar is a mobile information retrieval support and recommender system for supporting research activities. In the following list, the system requirements in each perspective are analyzed.

Research Support System

iScholar is a support system, specialized for research activities. The system shall understand a researcher's information needs and provide suitably personalized search results. Hoeber [40] noted "fundamentally, the personalization of Web search deals with the modeling of searchers' interests, and then using these models to affect the outcomes of their future Web search activities." Based on one's personal information such as one's academic interests and reading histories, the system shall provide filtered and re-ordered search results. To tackle the information overload problem, it is essential to filter the given information such as the whole retrieved data from a query or entire conferences/journals schedule.

Information Retrieval Support Systems

According to Yao [73], whereas IRS (Information Retrieval Systems) focuses on the search and browsing functionalities, IRSS (Information Retrieval Support Systems) targets more on supporting functionalities. The system shall show enhanced search results representation, as well as utilizing multiple databases for improved search quality. As mentioned above, the nature of information system research is diverse and inter-disciplinary. Hence, the system shall provide sources from multi-disciplines and multi-databases. Not only that, how to display the search results is no less important than the quality of retrieved data [40]. The system shall produce a representation of information that is easy to understand and simple enough to grasp at a glance [37].

Recommender Systems

For users, continuously accumulated digital-library contents could be overwhelming. The system shall analyze the interests and preferences of researchers and recommend personalized articles/conferences. Further, recent research has highlighted the importance of explanations of why certain items are recommended over others [14, 38, 57]. That could be achieved by providing a visual of personalized results [41] or being conversational between a system and user [63]. Herlocker et al. [38] advanced an opinion that meaningful explanations is a challenge which must be addressed to enable recommender systems to be more understandable, effective, and acceptable. In order to obtain credibility and satisfy users, the system shall convince users by providing not only reasonable recommendations but also valid reasons for those recommendations.

Mobile Systems

The goal is to develop a mobile application which serves as a useful tool while conducting research. It intends to shift research support systems from the Web to the mobile environment. The system shall not only take advantage of mobile technology, but also deal with its limitations such as capacity and usability.

Portable: always-on connection of mobile devices allow researchers to search information whenever they need [18]. More people are getting used to using smartphones everywhere and moreover for some people "smartphone only" experiences are on the rise. Evidence shows that the familiarity and portability of mobile devices can be enough supplement research tools, for instance, searching academic libraries and instant note-taking on the move [18, 27].

Maintainable: a system should adapt to changed requirements, and I believe this is especially true for support systems. Requirements for supporting an application domain may be uncertain at first or may change over time. Moreover, some research activities may not be ideally supported by mobile devices at present, yet they might become supported in the future. Mobile systems can be fixed and updated gradually in a cost-effective way. On one hand, bugs can be fixed if codes are readable and well-organized. On the other hand, existing features can be removed and new features can be added like blocks if codes are written in well-separated modules.

Personalized: being personal includes security issues, but at the same time offers an opportunity to have more information about the owner of a device [10]. Mobile devices know where the user is located, how fast the user is moving, current time of the day, and much more. Different personal information, mostly contextual data from mobile devices, may be used for different domains. In the domain of e-learning, context includes users' learning goals, background knowledge, motivations, and so on [?]. If personal information is adequately used, it might be possible to provide better support.

4.4 System Architecture

Architecture serves as a roadmap for the system building process by putting system components into the correct perspective and defining relationships among the components [59]. Since iScholar is an instance of MRSS (Mobile Research Support Systems) particularly for exploring/retrieving support, I first suggest the overall architecture of mobile research exploring/retrieving support systems architecture. iScholar's system design based upon the architecture is introduced in Section 4.4.2.

4.4.1 MRSS Architecture

Research activities have a large spectrum [43]. It is almost impossible to build a system that assists every aspect of research. Individual sub-systems design for each category - exploring/retrieving, reading, and writing support systems - will build up the entire MRSS framework. I propose the overall architecture of exploring/retrieving support systems (Figure 4.1), in other words literature review support systems.

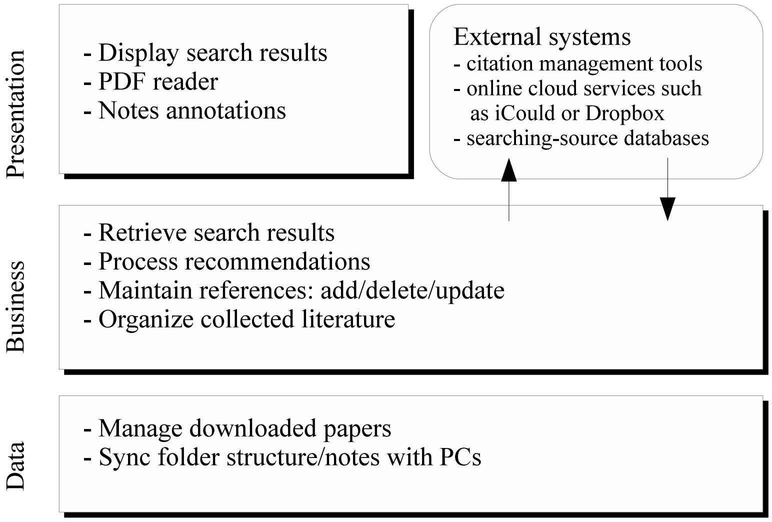

Presentation
- Display search results
- PDF reader
- Notes annotations
External systems
- citation management tools
- online cloud services such as iCould or Dropbox
- searching-source databases
Business
- Retrieve search results
- Process recommendations
- Maintain references: add/delete/update
- Organize collected literature
Data
- Manage downloaded papers
- Sync folder structure/notes with PCs

Figure 4.1: Architecture of literature review support systems

The system interacts with external systems such as citation management tools, online cloud services, and databases. The presentation layer is in charge of displaying search results of papers, and interacting with users. It may also act as a PDF reader including annotation feature. The business layer is responsible for communicating with the external systems, processing recommendations, and maintaining references. It works for retrieving search results, and organizing collected papers. The data layer is for managing actual file systems and synchronizing folders with PCs and clouds.

In general, the architecture serves as a system design for a literature review support system and sub-architecture of the whole MRSS framework. As stated, the architecture will serve as a roadmap for the iScholar system build process.

4.4.2 The System Architecture of iScholar

The iScholar proposed system design is for a mobile information retrieval support and recommender system that is supporting literature review activities. The system keeps researchers' information needs, and provides personalized search results. While traditional search systems give priority to automated searches and representations of the results, it is believed that better support systems must actively play a role at refining user queries, filtering returned results, and evaluating the relevance [40].

The proposed system will have both client and server parts since mobile devices are not yet powerful enough for all the required data processing. Mobile application architecture is determined by client type (thin/fat clients), server tier(s) (one-tier, two-tier, or three-tier), and connectivity (always, partial, or never) [49]. Table 4.1 lists the pros and cons of each selection. Among many possible combinations, for example 'zero-layer client, three-tier server, always connected' or 'three-layer client, three-tier server, partially connected', there are some considerations including [49]:

Table 4.1: Mobile application architectures - pros and cons (Source: [49])

		pros	cons
Clients	Thin	easy to maintain and support	servers require a high performance, client and server need to be connected all the time
	Fat	useful when connection between server and client are not guaranteed	heavily dependent to the client's operating system
Server	One tier	easy and quick to develop and deploy since three code layers exist on a single machine	low scalability, security
	Two tier	separating a database server from the presentation/application server - allows database server specialization	still low scalability, security, more expensive
	Three tier	scalable, allows database server specialization	most expensive and difficult to develop/manage

Conforming requirements: deciding connectivity is important because unlike desktops mobile devices are often disconnected [65]. For iScholar, server and clients do not have to be connected all the time since downloaded articles can be read off-line. Sometimes, however, they need to be connected when clients are sending a query or collecting data from the server. Therefore, 'partially connected' connectivity is required. A thin client has no custom code and relies entirely on server functions [49], so a client has to always be connected. Since the previous choice is 'partially connected', a thin client is not under consideration. I choose the 'fat client' type which has one or more layers of custom client code. Fat clients are used when a device is partially connected or the application uses device-specific features [49], in the iScholar case the device's local storage.

Performance/Scalability: it is expected that the processing overload of searching retrieval and recommendation evaluation is significant given relatively limited mobile systems' hardware capacity. I try to allocate as much work as possible to the server except personal information. In doing so, it is possible to minimize the client's workload and focus on the presentation part. Since there is no need to display anything on the server side, two-tier server layer is selected. Another advantage of this way is that future scalability becomes readily available. With no change of server-side code, simple implementations of user interface of client-side are all needed to extend to another mobile operating system.

It is worth pointing out too that iScholar performance has another important issue - it deals with at least a thousand search results for a query. Actual experiments on how long a search personalization would take under certain conditions has not been conducted, but this factor needs to be addressed for future study.

Security: security in mobile devices are more important than traditional systems

> since mobile users are often unaware of vulnerability and often socially engaged. The personal data that iScholar will utilize is users' name, publications list, academic interests, downloaded list, and reading history. The data is stored in the client side for the security reasons. However, such data needs to be transferred from time to time since personalization and recommendation modules are performed on the server side. Thus the challenge of having to protect sensitive data in transit as the application connects to some server component has to be addressed. This can be done using encryption such as SSL libraries that any mobile platform would offer.

All things considered, I choose the architecture of iScholar as 'two-tier server, three-layer client, and partially connected.' Figure 4.2 displays the overall system architecture. The server and client request/respond to each other intermittently. The client has three layers - presentation/business/data layers, and the server has two. The server maintains the conference data with a database, whereas the client stores the personal information into the system's file system.

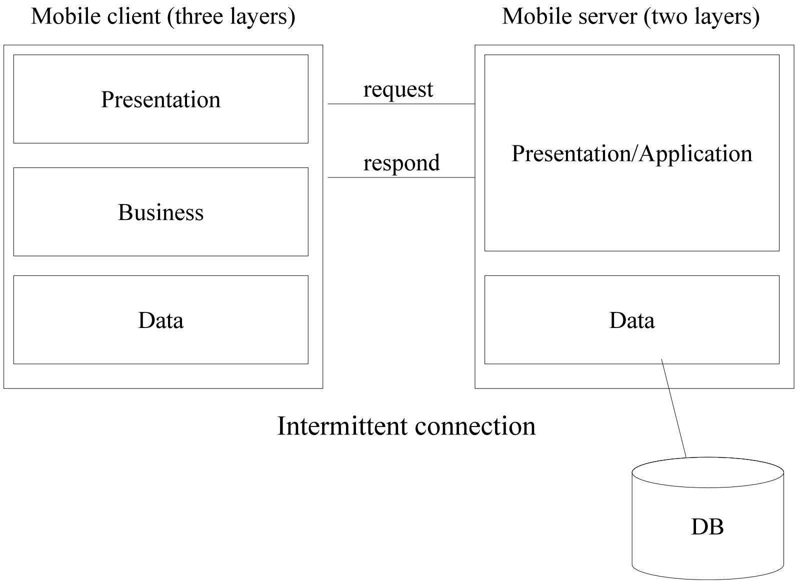


Figure 4.2: iScholar system architecture

4.4.3 System Components

Figure 4.3 shows four components of iScholar and the relationships between each of them. The search component is in charge of sending user queries to the server, returning search results to the client, and representing the returned data. If the results are matched to the articles that the user is looking for, then the user can move on to the article management part. Articles are bookmarked for future use or downloaded for off-line use in the article management component. Once an article is downloaded into local storage, annotations on important parts are possible. The annotations could be underlines, circles, or text comments.

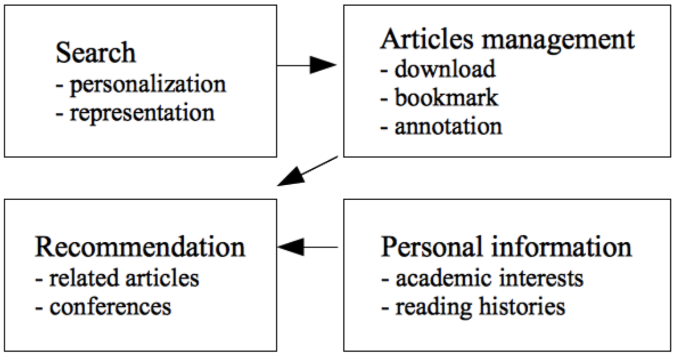


Figure 4.3: iScholar system components

Basic information of individuals could be entered and reading histories are recorded either automatically or manually. The personal information component stores such data, processes and passes it when requested from other components. When enough personal data is gathered, the recommendation component matches with conference and articles database from the server. Personalized and customized recommendations are generated from the component. The arrows show the relationships between components. For example, personal information component maintains the academic interests and reading histories of a user, and the system recommends related articles and conferences based on the personal information.

Figure 4.4 details the task allocations of sub-components to server/client sides and draws the overall structure of the system prototype. Each sub-component is divided into modules and is placed where it should be in the layered architecture.

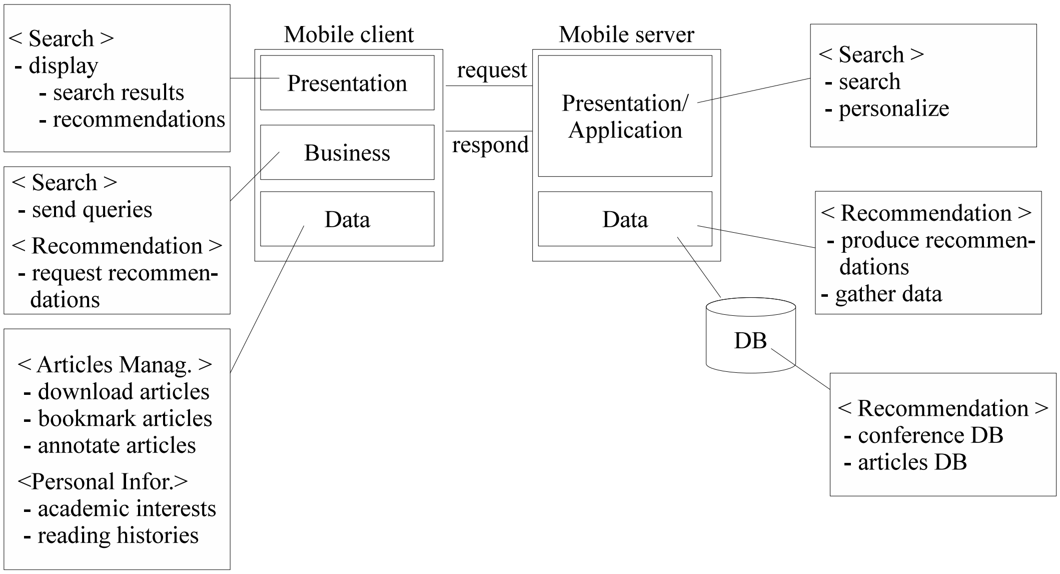


Figure 4.4: iScholar system sub-components

4.5 System Design

Designing a software system entails good understanding of the domain, application of relevant knowledge, consideration of alternatives, and evaluation of the alternatives [59]. The overall iScholar system design, user profile modeling, and database design are discussed in this section.

4.5.1 MVC Design Pattern

The Cocoa Touch framework underlying iOS applications is based on Model-View-Controller (MVC) design pattern, which is useful for separating the program logic from user interfaces [2]. MVC is central to the design of any iOS application [44].

Benefits are that objects are more reusable, interfaces are better defined, and applications are more extensible [10, 44]. It divides a system design into three roles [2]: Model, View, and Controller. Figure 4.5 describes the client-side's MVC components of each role.

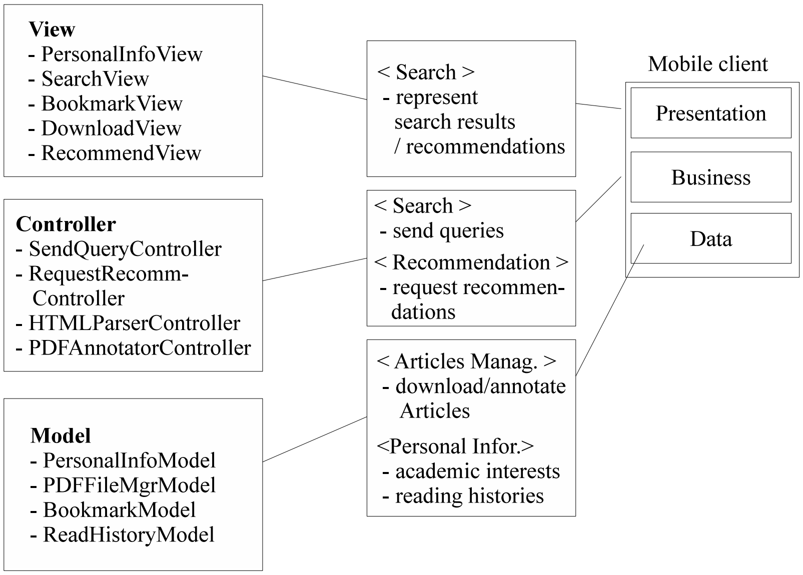


Figure 4.5: MVC design of iScholar client

Model objects - Model objects hold application data and define the data logic. A model object should not be connected with the user interface and presentation in general [44]. A common element of model objects is persistent data [44].

PersonalInfoModel: Any personal information gathered by the system is stored to this object. The data can be collected explicitly and implicitly. For instance, a user's academic interests could be extracted from the user's publications list (implicitly). On the other hand, the system asks a user to enter full name and his/her publications list (explicitly).

PDFFileMgrModel: Once a user selects to download a paper, the system stores the paper into the PDF file format. Annotations are stored the original PDF files. Each PDF page is treated as a frame with annotations and annotated frames are combined as a PDF file. The object maintains the PDF file list and returns the list whenever other objects request it.

BookmarkModel: This object represents each bookmark that a user created. Each bookmark has information about the title, author(s), year, and link (if exists) of a paper.

ReadHistoryModel: Every time a user opens a paper online or downloaded paper, the system logs the reading history. These logs are used to analyze the user's interests and generate recommendations papers to read.

View objects - These objects are responsible for displaying and interacting with users.

PersonalInfoView: It displays a person's information, such as name, publications list, academic interests, and bookmark/downloaded papers.

SearchView: Users enter search terms that the users wish to search among scholarly literature. The view sends a request to SendQueryController and displays the returned results.

BookmarkView: This object displays the publications list that a user bookmarked.

DownloadView: The downloaded papers list is displayed. If a user clicks a title of the list, it opens the PDF viewer and PDFAnnotatorController.

RecommendView: The object sends requests to RequestRecommController and displays the returned recommendations list.

Controller objects - Controllers act as an intermediary between view and model objects [44]. They are often in charge of passing data to view objects they need to display from model objects.

SendQueryController: Once this object receives a search request, it sends a query to

Google Scholar and retrieves the results. After all it returns the results to the view that requested it.

RequestRecommController: Actual recommendation process is performed on the server side. This object on the client side is in charge of sending a request to the server and passing the results to a view.

HTMLParserController: As the search results from Google Scholar are in HTML format, the system needs a controller that parses the HTML page. The object parses the page and returns the result with the format that other objects understand.

PDFAnnotatorController: The object creates, edits, and displays PDF annotations on top of the downloaded PDF files. Annotated PDF files will be stored separately with the original ones.

In practice, often two different objects are merged, which is called a view-controller or model-controller [44]. When both a view and controller are concerned only with each other, combining their roles in one object is acceptable [44]. It helps to reduce interactions between objects. For example, PersonalInfoView displays user profile only and PersonalInfoController mostly concerns retrieving the profile data from the model object. Then for convenience's sake two object are combined to PersonalInfoViewController. These combined view-controllers of iScholar will be discussed afterwards in Chapter 5.

4.5.2 User Profile Modeling

In iScholar a researcher is modeled as a combination of his/her academic interests, publications, and reading history. Publications list is entered by a researcher, the researcher's academic interests are extracted from the publication list, and the reading history is gathered by the system every-time a researcher reads a paper. Each component is discussed below in turn.

Academic interests

iScholar automatically extracts a user's academic interests from the user's publications list. It is assumed that if particular terms appear in the user's papers frequently it is likely that the terms represent the user's academic interests. To decide which terms are showing repeatedly, the term frequency (number of times a term occurs in a document [55]) is calculated. In information retrieval, the simplest approach to compute the term frequency is to count the number of occurrences of a term t in a document d and assign equal weight to each term t, denoted $tf_{t,d}$ [55].

On the other hand, it is notable that there are certain terms that have little relevance to discriminate between documents [55]. For instance, papers about mobile computing are likely to have the terms *mobile* and *computing* in almost every paper. To distinguish these terms that occur too often, but are not very meaningful for discrimination of documents, commonly inverse document frequency (idf) is used [55]:

$$idf_t = log\frac{N}{df_t}, \tag{4.1}$$

where df_t is the document frequency of a term, i.e., number of documents in the entire documents that contain a term t, and N is the total number of the entire documents.

By combining the term frequency and inverse document frequency, a composite weight of each term in each document (id-tdf) is produced [55]. To calculate the td-idf, the below equation is used:

$$tf\text{-}idf_{t,d} = tf_{t,d} \times idf_t. \tag{4.2}$$

Tf-idf is high if a term t appears many times in a document but not many documents in the entire documents [55]. Td-idf is low if a term t appears few times in a document or appears in virtually all documents [55].

Algorithm 1 explains how iScholar estimates a user's academic interests from the user's publications list using the tf-idf. The algorithm gets inputs of a researcher's publications list and returns the researcher's extracted academic interests. For each text from a user's publications list, an array or words is extracted and the td-idf of each word is calculated. In Line 6 of the algorithm, unnecessary words that are not adequate to be a keyword for one's research such as an, the, or we, in other words stop words, are removed. The reason is that stop words have high frequencies, therefore almost only stop words will be left if they are not removed [69]. Once the number of occurrences are calculated, the tf scores are adjusted based on publication years. The publication years of papers are considered since one's academic interests change over time [66]. The weight of year factor is higher if a paper was published recently. Another weight factor is the location of each term. Being located in a title or abstract reflects the importance of terms. If a term is located in a paper's title, the term gets the highest weigh. If a term is located in abstract or conclusion the weight for the term is the second highest. The reason is that an author will try to make his title, abstract, and conclusion include all the relevant terms that he/she believes.

The academic interests are inferred based on the calculated td-idf scores. Top-ten td-idf highest terms are assumed to be the researcher's academic interests. Created words list and calculated td-idfs are not stored since extracting academic interests happens only once when the application is launched for the first time. At present the case that one's publications list is updated is not considered for the simplicity. Future development that allows updating user profiles needs to be supported. Another consideration is that normally an academic interest consists of compound words (e.g., rough sets or computational geometry) and Algorithm 1 only considers frequency of one word each. Getting the frequency of compound words would be a challenging issue for future improvement. Using existing statistical analysis tools of input contents, such as Yahoo Content Analysis[31] or Google BigQuery services[32], would be a feasible alternative.

Algorithm 1 Building User Profile - Extracting Academic Interests

```
procedure EXTRACT_ACADEMIC_INTEREST(publications_list)
    empty $academic\_interests$
    convert the publications list PDFs to text format
    for each text do
        split the text into an array of words
        remove stop words (‘the’, ‘an’, ‘we’)
        for each word do
            $tf_{t,d}$ = number of occurrence                          ▷ calculate tf
            $tf_{t,d,y}$ = tf with publication year          ▷ adjust tf based on year published
            $tf_{t,d,y,l}$ = tf with location of the term          ▷ adjust tf based on location
            $idf_t = log\frac{N}{df_t}$                                ▷ calculate idf
            $tf\text{-}idf_{t,d,y,l} = tf_{t,d,y,l} \times idf_t$                ▷ calculate tf-idf
        end for
    end for
    sort terms by the tf-idf scores, highest first
    insert the top ten terms to the $academic\_interests$
    return $academic\_interests$
end procedure
```

These academic interests are used extensively in many iScholar modules that will be discussed later.

Past works

It is intuitive that a researcher's publications list reflects the researcher's past research interests [66]. Information from a researcher's past papers' references and other papers that cited the researcher's paper can be modeled and integrated to iScholar's user profile. Past works are a part of the data that iScholar integrates into recommendation processes. For getting the information, iScholar adapts Kazunari and Min-Yen [66]'s proposed method that constructs user's interests profile derived from the past publications list. For each paper p of a researcher's publications list, p is transformed into a feature vector f_p [66]:

$$f_p = (w_{t1}^p, w_{t2}^p, ..., w_{tm}^p), \tag{4.3}$$

where m is the number of keywords (distinct terms) and $t_k(k = 1, 2, ...m)$ refers to each keyword. Using term frequency, each element w_{tk}^p of f_p are defined as follows [66]:

$$w_{tk}^p = \frac{tf(t_k, p)}{\sum_{s=1}^{m} tf(t_s, p)}, \tag{4.4}$$

where $tf(t_k, p)$ is the frequency of term t_k in a paper p. In this Kazunari and Min-Yen [66] method, term frequency is preferred over tf-idf because the number of a user's publications list tends to be small.

Based on the set of feature vectors f_p, a user's profile regarding his/her past publications is created using the relations between each published paper and its citation/referenced papers. To clarify, citation papers are the papers that the paper is citing and referenced papers are the papers that are referencing the paper. Let $W^{p_u->v}$ be the multiplicative coefficient where citation paper is u and referenced paper is v. Weights are assigned to apply the influence of citation/referenced papers. Those who are interested in the three different weighting schemes can find them in Kazunari and Min-Yen [66]'s original paper. The *linear combination* weighting scheme ($W^{p_u->v} = 1$) is used in iScholar for simplicity. The scheme treats each neighbouring paper v the same with the researcher's own paper u.

Let us say a researcher has published papers $p_i(i = 1, ..., n)$. Each paper p_i may cite other papers: $p_{i->ref_y}(y = 1, ..., l)$, and it also may be cited by other papers: $p_{c_x->p_i}(x = 1, ..., k)$. The feature vector F^{p_i} for each paper p_i are constructed, using its citation papers $f^{p_{c_x}->p_i}$, their weights $W^{p_{c_x}->p_i}$, referenced papers $f^{p_i->ref_y}$, their weight $W^{p_i->ref_y}$:

$$F^{p_i} = f^{p_i} + \sum_{x=1}^{k} W^{p_{c_x}->p_i} f^{p_{c_x}->p_i} + \sum_{y=1}^{l} W^{p_i->ref_y} f^{p_i->ref_y}. \tag{4.5}$$

Finally, using equation 4.5, the user profile P_{user} is constructed:

$$\begin{aligned} P_{user} &= W^{p_{n->1}} F^{p_1} + W^{p_{n->2}} F^{p_2} + ... + W^{p_{n->n}} F^{p_n} \\ &= \sum_{z=1}^{n-1} W^{p_{n->z}} F^{p_z} + F^{p_n}, \end{aligned} \tag{4.6}$$

where $W^{p_{n->z}}$ refers to each weight assigned to paper $p_{n->z}$.

The feature vector of each candidate paper is constructed. Then the cosine similarity between the user profile and the constructed feature vector is calculated as follows [66]:

$$sim(P_{user}, F^{Prec}) = \frac{P_{user} \cdot F^{Prec}}{|P_{user}| \cdot |F^{Prec}|}. \tag{4.7}$$

Algorithm 2 presents how candidate papers are compared with a researcher's constructed user profile and so how certain papers are selected to recommend.

Algorithm 2 Recommendation - Selection of Papers Relevant to User Profile

```
procedure RECOMMEND_PAPERS(constructed_user_profile, candidate_papers)
    for each candidate paper do
        $W_{t_k}^{Prec} = \frac{tf(t_k, Prec)}{\sum_{s=1}^{m} tf(t_s, Prec)} \cdot log \frac{N}{df(t_k)}$    ▷ use td-idf
            ▷ $tf(t_k, Prec)$ is the term frequency of term $t_k$
            ▷ $N$ is total number of papers to recommend
            ▷ $df(t_k)$ is the number of papers in which the term $t_k$ appears

        $f^{Prec} = (W_{t1}^{Prec}, W_{t2}^{Prec}, ..., W_{tm}^{Prec})$
            ▷ calculate the feature vector
            ▷ m is the number of distinct terms, $t_k$ is each term

        $F^{Prec} = f^{Prec} + \sum_{x=1}^{k} W^{p_{c_x} -> Prec} f^{p_{c_x} -> Prec} + \sum_{y=1}^{l} W^{Prec -> ref_y} f^{Prec -> ref_y}$

        $sim(P_{user}, F^{Prec}) = \frac{P_{user} \cdot F^{Prec}}{|P_{user}| \cdot |F^{Prec}|}$
            ▷ compute similarity between the $P_{user}$ and $F^{Prec}$
    end for
    rank the set of candidate papers in decreasing order
    return the top ten papers
end procedure
```

The algorithm returns top ten similar papers to a user profile. The iScholar client then shows the recommendations to the user.

Reading history

The other source that iScholar integrates into recommendation processes with past works is the reading history of a user. It is assumed that the more times a researcher reads a paper the more the researcher is interested in the paper. So papers that are similar with the papers in the user's reading history are likely to be close to what the user wants. Every time a user clicks a paper (so assuming reads it) the system logs the title of the paper. If a paper is read by a user more than five times, the paper goes to the user's *frequently read* list. Once a user asks the system to recommend papers to read, the system considers the *frequently read* list and recommends similar papers with the list. The list is used by the personalize module in the server when it decides how a candidate paper for a recommendation is similar to the user profile(refer Algorithm 3).

4.5.3 Database Design

iScholar uses Google Scholar as the external publications database, but it maintains its own conferences database. This section explains the databased design - selection of the database sources, data representation, and gathering the data.

Publications database - Google Scholar

One of main objectives of iScholar is to provide multidisciplinary searches and a full-text of articles. According to Google Scholar^{33}, their coverage includes conference papers, theses and dissertations, books, pre-prints, and other scholarly literature from broad areas of research. By utilizing Google Scholar, iScholar can provide the same broad coverage. One other big advantage is that Google Scholar provides links to full-text as long as it can. Unless a student or scholar has individual subscriptions for journals, they mostly access the journal papers only when they are within a university or library that has a subscription for the journals. Google Scholar gives freedom from such limitation. Google Scholar thus fulfills, in large part, the requirement of iScholar.

One consideration is that there is little concrete information about how Google Scholar ranks publications. One study [7] suggested that Google Scholar's ranking algorithm focuses on citation counts based on the fact older articles are often retrieved in top positions. However, how Google Scholar ranks publication and which factors Google Scholar treats more importantly is still relatively unknown. Being uncertain at the database's ranking algorithm could affect the system's search logics.

The other consideration is that not all links to full-text are available through Google Scholar. If Google Scholar has a direct link to a paper it displays the link along with where the paper is physically located, for example “psu.edu” or “uregina.ca.” If it does not have a link to a certain paper it displays another link that might take users to a publisher’s website which might need a subscription. Suggestion of Google Scholar for this matter is to click on a link labelled “All versions” under the search result and try alternative sources. At this moment only one depth searching will be performed for simplicity but in the future availability of full-texts might be improved by applying multi-depth searching as suggested.

Conferences database - WikiCFP

The aim is to automatically gather conferences information by facilitating one of the existing services. There are online directory services that manually store data for conferences such as confsearch[34], Conference Alerts[35], Microsoft CFP Calendar[36], and WikiCFP[37]. Since individual organizational services, e.g., IEEE conferences or ACM conferences, provide only their own publications, they are not under consideration. The following briefly introduces each service and discusses the availability.

Confsearch: Confsearch provides searching functionality and additional analysis such as graphical dates representation and ratings. Since it is based on DBLP (Computer Science Bibliography[38]) data for conferences, the domain is limited to computer science area, and some additional information, for instance website URL, start and end dates, etc, might be missing unless a user manually modifies the meta-data.

Conference Alerts: As the name suggests, it is mainly used for getting upcoming interested-events alerts. Anyone who has an organizer account can update an event to the database, and a user can subscribe to an alert by keyword or place. The database is maintained by organizers who wish to promote their conferences.

Microsoft CFP Calendar: Microsoft visualizes calls for paper announcements. CFP calendar makes it easy to check submission deadline, acceptance notification dates, and duration of the conferences in an organized way. It also provides two ways of visualization, namely timeline-view and map-view. Microsoft provides academic search API for publications, but unfortunately not for the calls for paper data.

WikiCFP: WikiCFP has been a major portal for academic paper submission deadlines in the computer science community [36]. It is a semantic wiki for calls for papers. According to the website, CFP data is released free for academic purposes and crawling is also permitted as long as each query is issued at more than five seconds interval.

First, iScholar does not need past data for conferences because iScholar recommends a conference that one can possibly submit to or attend. Therefore it only uses upcoming ones. Besides, commercial databases or databases that are hard to crawl because of visualization are ruled out. With all these considerations, WikiCFP is selected as iScholar's conference database. How to gather and compile the data will be explained immediately below.

Conferences database crawler

Database crawling module is located in the server-side of iScholar. It periodically gathers and saves the data for conferences. The module decodes target webpage(s) (WikiCFP), identifies new data for conferences, and stores it into its own conference database. Implementation detail of this module can be found at Section 5.2.

In the future, rather than implementing the whole web-crawler from scratch, using a pluggable open source crawler, Nutch[39], is proposed [36]. It is a high-performance, text search engine library [36].

Conferences data representation

A CFP (Call For Paper) column is composed of seven parts. They are title, link, date, place, deadline, abstract, and category. Table 4.2 lists the attributes and real-world examples. *Title* and *category* attributes are used to estimate the similarity between the current working paper and upcoming conferences. If the similarity is high enough, the conference is recommended to submit.

Table 4.2: A CFP data representation

Attribute	Description	Example
Title	name of the conference	RecSys 2013 : ACM International Conference on Recommender
Link	webpage of the conference	http://recsys.hosting.acm.org/recsys13/
date	when the conferece happens	Oct 12, 2013 - Oct 16, 2013
Place	where the conference takes place	HONG KONG
Deadline	submission deadline of the conference	TBD
Abstract		We are pleased to invite you to participate in the premier annual event on research and applications of recommendation technologies, the Seventh ACM Conference on Recommender Systems (RecSys 2013). The previous conferences have been distinguished by a strong level of interaction between researchers and practitioners in the sharing of ideas, problems and solutions, and the 2013 conference will continue in this tradition. The fully-refereed proceedings will be published by the ACM and, like past RecSys proceedings, are expected to be widely read and cited...
Category		recommender systems data mining IR user modeling

Chapter 5

iScholar System Implementation

Implementation of the designed system demonstrates the feasibility of the anlayzed functionalities and developed design [59]. The final goal of this process is to gain insights into the concepts, framework, and design alternatives [59]. Why specific techniques are chosen over other platforms, databases, or languages is discussed in Appendix A.

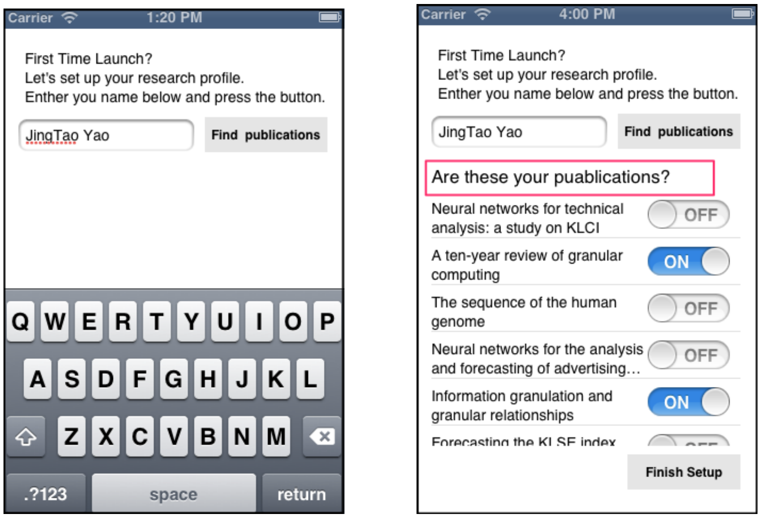
5.1 System Prototype - The Client side

iScholar client application has four model objects and five view-controllers objects. The detail of each object is introduced in this section.

5.1.1 Model Objects

iScholarPersonalInfo Class: The class holds a user's personal information such as name, publications list, academic interests, downloaded papers, and reading history. When the system is launched for the first time, it asks for the user's name - Figure 5.1, (a). Then the system searches publications by the name from Google Scholar and makes sure that the returned publications are actually the user's publications.

The user can select *on* button(s) - Figure 5.1, (b) - if the publications are his/hers. The system stores the entered user name and the publications list permanently at this point. Using the publications list, the user's academic interests and past works are constructed as explained in Section 4.5.2. This step is performed only once at the first time.



First Time Launch?
Let's set up your research profile.
Enther you name below and press the button.

JingTao Yao

First Time Launch?
Let's set up your research profile.
Enther you name below and press the button.

JingTao Yao

Are these your puablications?

Neural networks for technical analysis: a study on KLCI	OFF
A ten-year review of granular computing	ON
The sequence of the human genome	OFF
Neural networks for the analysis and forecasting of advertising…	OFF
Information granulation and granular relationships	ON
Forecasting the KLSE index	OFF

(a) Capture 1

(b) Capture 2

Figure 5.1: iScholar - first time launch

Table 5.1 lists the *PersonalInfoModel* class properties and actual examples.

Table 5.1: PersonalInfoModel class properties

Property	Data Type	Example
Name	NSString	‘JingTao Yao’
Publiations	NSMutableArray	‘Granular Computing: Perspectives and Ch..’, ‘Rule Measures Tradeoff Using Game-theoretic..’, ...
Interests	NSMutableArray	‘granular computing’, ‘rough sets’, ‘soft computing’, ...
Downloads	NSMutableArray	‘Equilibrium points in n-person games’, ‘Granular computing: basic issues and possible solutions’, ...
Reading History	NSMutableArray	‘Granular computing: an emerging paradigm’, ‘Web-based customer decision support systems’, ...
Link (if exists)	NSURL	‘http://scholar.google.ca/citations?user =WHpY3ogAAAAJ&hl=en&oi=ao’

iScholarPDFFileMgr Class: This class manages the downloaded files list. It retrieves the whole list or a particular file when requested. Table 5.2 shows the class description.

Table 5.2: iScholarPDFFileMgr class design

	Design	Description
Attributes	*private* string contentspath[128] *private* integer count	“/Users/downloads/” how many PDF files are stored
Operations	*public* array getDownloadsList() *public* array getSearchHistoriesList() *public* file *getaPDFfile	returns downloads list returns SearchHistories list returns a PDF file requested
Responsibility	services various operations related to file system	

iOS for iPhone applications does not provide the direct method to draw annotations (lines, circles or texts) on top of PDFs. iScholar takes these steps to resolve: 1) opens PDF each, not the whole file, 2) gets frame information from the page, which the system can manipulate freely, 3) receives user inputs - lines, circles or texts. Every time a user enters new input, the system re-draws new frame, and 4) stores each frame itself and makes up the whole document.

iScholarBookmark Class: The articles can be bookmarked for later access. Each bookmark data has below structure (Table 5.3). The *Number* attribute (how many times a paper has been read after it was bookmarked) is stored for later use in reading history module.

Attribute	Description	Type	Size
Title	title of the paper	*string*	512
Date	when this bookmark was created	*string*	128
Number	how many times the paper has been read (opened)	*int*	4

Table 5.3: iScholarBookmark class attributes

iScholarReadHistory Class: The logical structure of a single data of each history is given below (Table 5.4). Necessary minimum attributes for this class are date, number, title, and download url.

Attribute	Description	Type	Size
Date	most recent time when the paper is opened	*string*	128
Number	how many times the paper has been read (opened)	*int*	4
Title	title of the paper	*string*	512
Download url	physical address of the paper	*string*	1024

Table 5.4: iScholarReadHistory class attributes

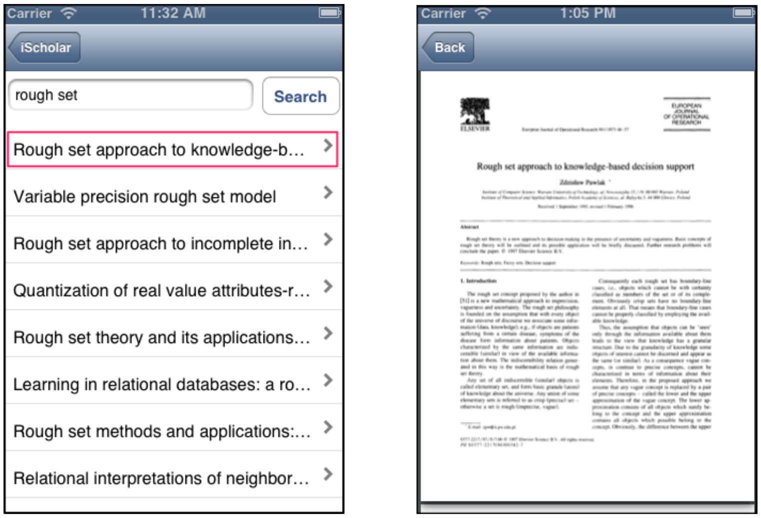
5.1.2 ViewController Objects

Model objects are implemented as designed in Section 4.5.1. It is acceptable to combine views and controllers if they concerns each others only, called view-controllers [44]. iScholar has five viewControllers for efficient code management instead of five views and five controllers.

iScholarSearchViewController Class: This class mainly interacts with the server's search module. As seen in Figure 5.2, (a) below, it sends the user entered search term to the server. The server returns the most relevant papers with the search term. Then the class adds personalization to them by sending them to the personalize module (Algorithm 3). Eventually the system displays the final results. Once a user selects a paper in the list, the system opens up the PDF viewer and displays the content of the paper (Figure 5.2, (b)). For the present the system displays the searching results without personalization. Algorithm 3 proposes how the system personalizes the returned search results.

Algorithm 3 Search Results Personalization

```
procedure PERSONALIZE_SEARCH_RESULTS(returned_search_results, user_profile)
    for each returned search result from the server do
        download the PDF file if available
        build the feature vector (refer Section 4.5.2)
        set similarity to 0
        for each publications list and reading history do
            Similarity += SIM(paper, publications/history)
            adjust the search results order based on the similarity
        end for
    end for
end procedure
```



rough set

Rough set approach to knowledge-b...

Variable precision rough set model

Rough set approach to incomplete in...

Quantization of real value attributes-r...

Rough set theory and its applications...

Learning in relational databases: a ro...

Rough set methods and applications:...

Relational interpretations of neighbor...

Rough set approach to knowledge-based decision support

(a) Capture 3 (b) Capture 4

Figure 5.2: iScholar - search view controller

Once searching results are returned from Google Scholar, the system considers the similarity between the user and returned results and adjusts the ranking of the returned searching results. It is assumed that adjusted list is likely to interest the user more because the items are *similar* to the user.

iScholarProfileViewController Class: As mentioned earlier, a researcher's name and publications list are given by user inputs and academic interests are extracted automatically from the publications list. Figure 5.3 demonstrates the sequence of user profile/academic interests building processes and how they are used in the process. The constructed user profile is shared with the server's recommend module. Whenever the user profile is needed the recommend module requests it to the *iScholarPersonalInfo* model class. Users can find their information under "My Profile" menu of the application (Figure 5.4, (a)). A researcher's name, publications list, and academic interests are displayed (Figure 5.4, (b)).

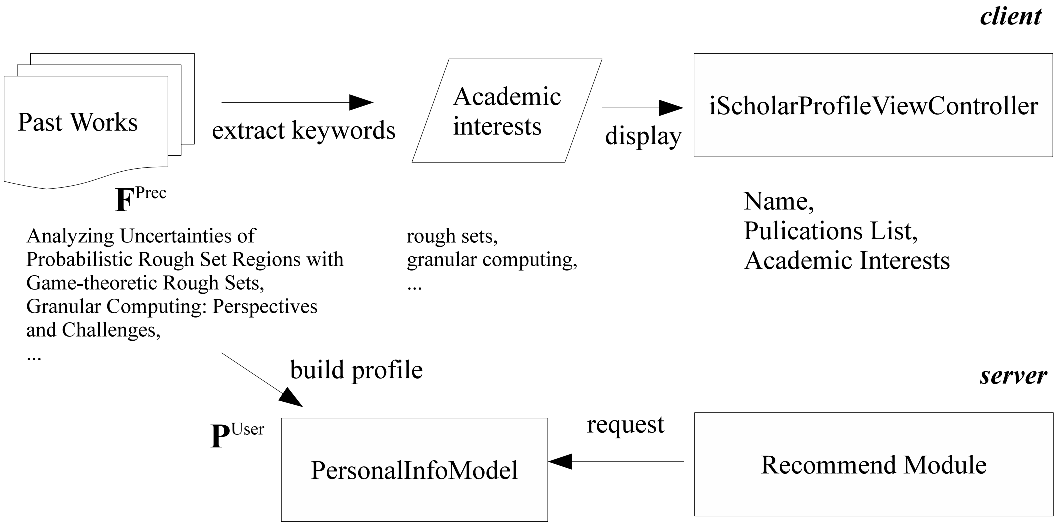
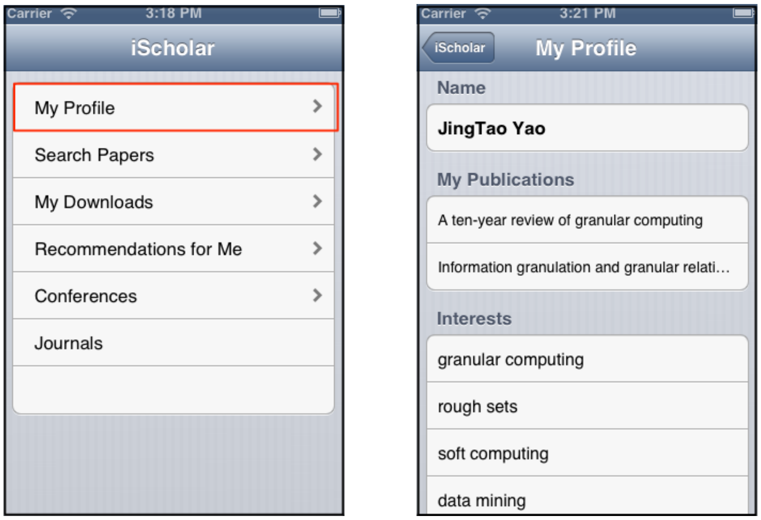

client
Past Works
extract keywords
Academic interests
display
iScholarProfileViewController
$\mathbf{F}^{Prec}$
Analyzing Uncertainties of Probabilistic Rough Set Regions with Game-theoretic Rough Sets, Granular Computing: Perspectives and Challenges, ...
rough sets, granular computing, ...
Name, Pulications List, Academic Interests
build profile
server
$\mathbf{P}^{User}$
PersonalInfoModel
request
Recommend Module

Figure 5.3: iScholar user profile



Name

JingTao Yao

My Publications

A ten-year review of granular computing

Information granulation and granular relati...

Interests

granular computing

rough sets

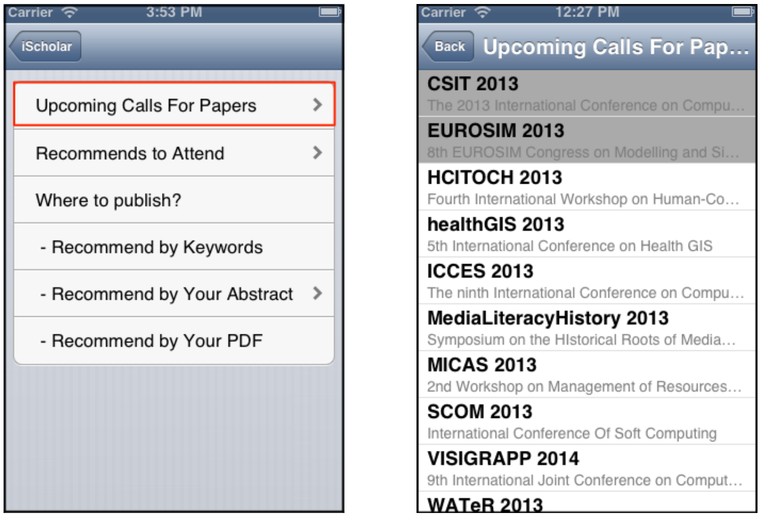
soft computing

data mining

(a) Capture 5 (b) Capture 6

Figure 5.4: iScholar - profile view controller

iScholarConfUpcomingViewController Class: It sends a request to the server (particularly the DBManager module), retrieves the latest upcoming calls, and displays the list (Figure 5.5, (a)). Recently added calls for paper are highlighted by a different color (Figure 5.5, (b)). Once each call for paper is selected, detailed information is shown such as the conference's website, date and place, submission deadline, and so on.



CSIT 2013
The 2013 International Conference on Compu…
EUROSIM 2013
8th EUROSIM Congress on Modelling and Si…
HCITOCH 2013
Fourth International Workshop on Human-Co…
healthGIS 2013
5th International Conference on Health GIS
ICCES 2013
The ninth International Conference on Compu…
MediaLiteracyHistory 2013
Symposium on the HIstorical Roots of Media…
MICAS 2013
2nd Workshop on Management of Resources…
SCOM 2013
International Conference Of Soft Computing
VISIGRAPP 2014
9th International Joint Conference on Comput…
WATeR 2013

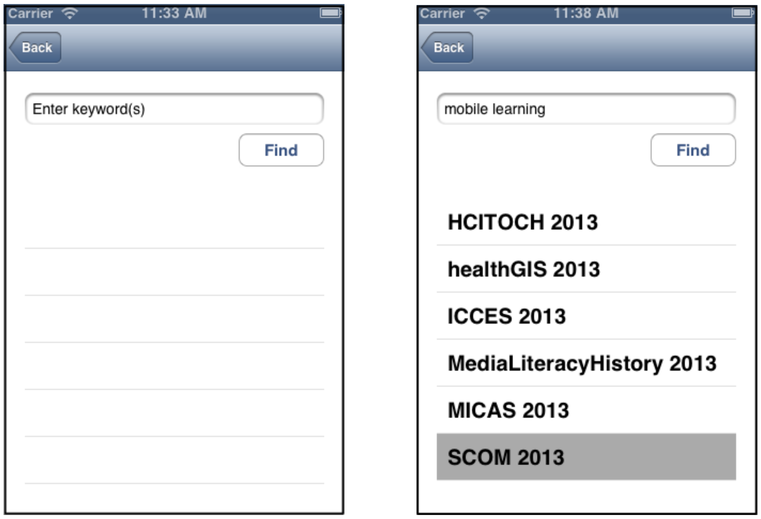
(a) Capture 7 (b) Capture 8

Figure 5.5: iScholar - iScholar upcoming conferences view controller

iScholoarRecommendConferencesViewController Class: Conferences are recommended by three different criteria - keywords, abstract, and the entire file that one has been writing. The writing does not have to happen in iScholar. One can copy his/her abstract from another word or PDF file and paste it to the view- controller.

Recommend by Keywords

The most basic scheme is recommendations by keyword(s) (Figure 5.6, (a)). As an example, if a user wants to search conference(s) that are related to *mobile learning* so the keyword are entered by the user, the client asks to the server and the server looks up the conferences database. Then the server returns the most suitable upcoming conferences and the returned list is displayed by the client (Figure 5.6, (b)). The actual selection process of conferences is carried out by the server (refer Section 5.2.4 - recommender module).



mobile learning

HCITOCH 2013
healthGIS 2013
ICCES 2013
MediaLiteracyHistory 2013
MICAS 2013
SCOM 2013

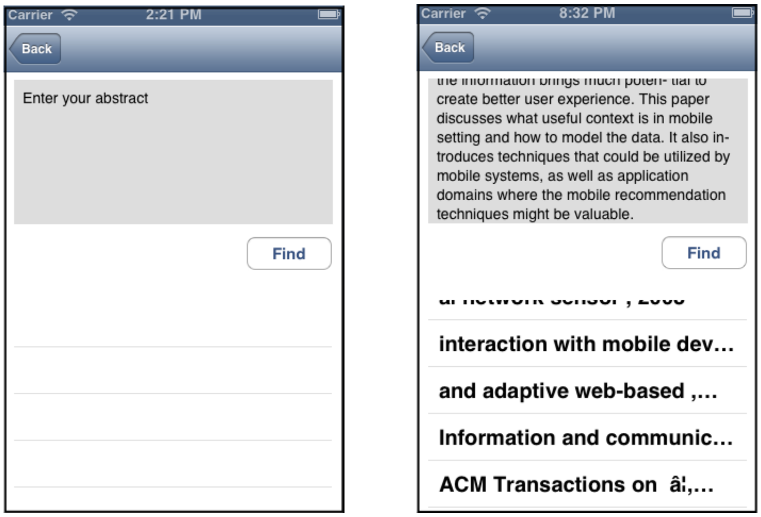
(a) Capture 9

(b) Capture 10

Figure 5.6: iScholar - iScholar recommend by keyword view controller

Recommend by abstract

An abstract is a paragraph that summaries a research article, so it naturally includes important words of a paper. This scheme is somewhat similar to the "recommendation by keywords" but gives more sense to users by not missing important word(s) of a paper since it gets a paragraph for an input. There might be chance to produce more accurate recommendations than simply using keyword(s) for finding relevant conferences.



the information brings much poten- tial to create better user experience. This paper discusses what useful context is in mobile setting and how to model the data. It also in- troduces techniques that could be utilized by mobile systems, as well as application domains where the mobile recommendation techniques might be valuable.

interaction with mobile dev…

and adaptive web-based ,…

Information and communic…

ACM Transactions on â¦,…

(a) Capture 11   (b) Capture 12

Figure 5.7: iScholar - iScholar recommend by abstract view controller

Recommend by PDF

At present, it is not technically viable for an iPhone application users to upload a file to the application. For future extension, it is possible to get an input file from a user, compare the file with upcoming conferences, and recommend the most relevant conferences to submit.

iScholoarRecommendJournalsViewController Class: Journals data is rather different from conferences database in terms of the frequency. Journals are published regularly, mostly monthly or biweekly, while conference proceedings are issued usually once a year. Even though, characters and academic interests of journals are not changed frequently. Therefore, the journal data does not need to be updated once a database is gathered. Because of the reasons, two separate databases and accordingly two different view-controllers are designed.

The user scenario for journal recommendations is similar to the conference recommendations. Once a user wishes to find a journal to submit, the system run through the same process only with the journal database and displays recommended journals.

5.2 System Prototype - The Server Side

Figure 5.8 presents the overall structure of the server-side prototype. Implemented source codes for some modules can be found at Appendix A.2. The data layer in the server side performs tasks associated to databases (DBManager module). The presentation/application layers contain search/personalize/recommender/crawling modules.

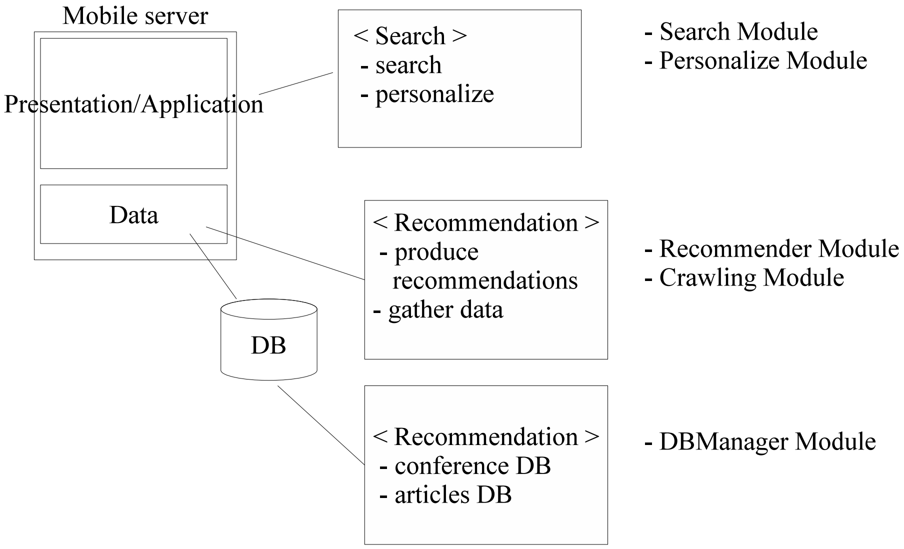


Figure 5.8: System prototype - server side

5.2.1 Search Module

The following sequence diagram elaborates on the interactions between the client, server, and external sides in the searching process. The client sends a query term, and then the search module passes the query to Google Scholar. Since Google Scholar does not provide APIs to developers, iScholar uses a detour. *Beautiful Soup*[40] library is used to get the 'results html-page' from Google Scholar and parse the html-page. Beautiful Soup is a Python HTML/XML parser designed for quick turnaround projects such as screen-scraping. The search module extracts title, url, author, abstract, etc. It then sends extracted data back to the client side. Listing B.1 shows the source code of this module.

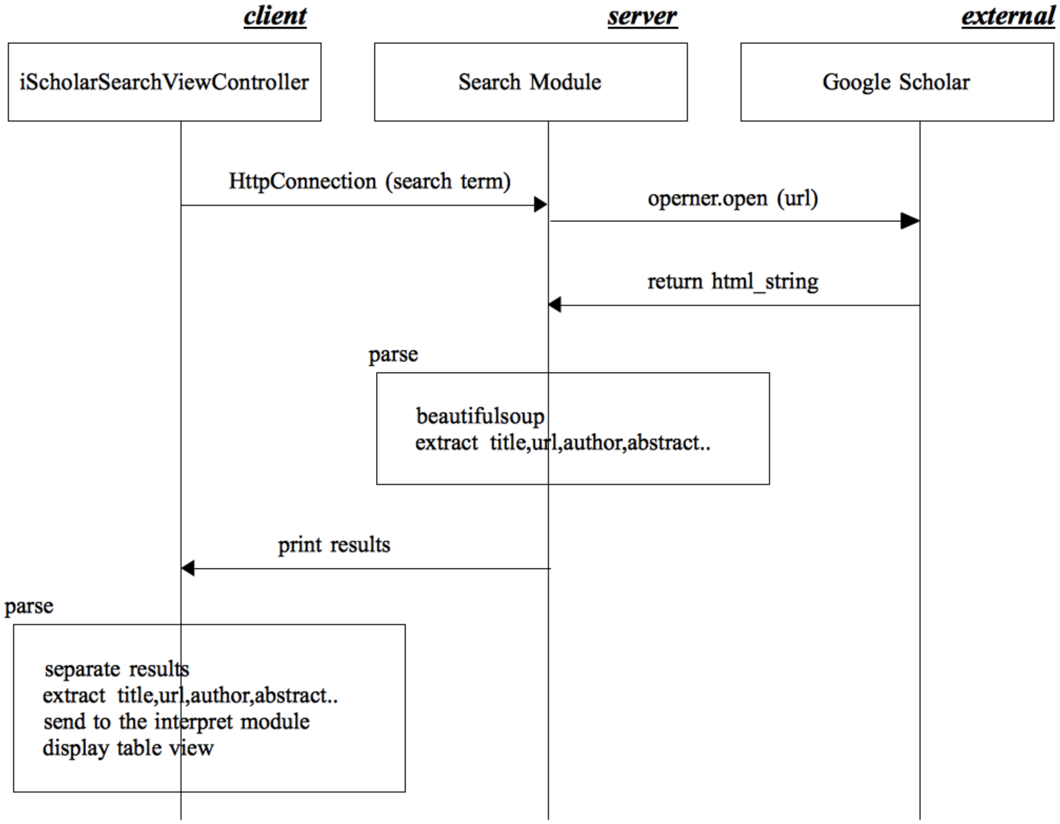

client
server
external
iScholarSearchViewController
Search Module
Google Scholar
HttpConnection (search term)
operner.open (url)
return html_string
parse
beautifulsoup
extract title,url,author,abstract..
print results
parse
separate results
extract title,url,author,abstract..
send to the interpret module
display table view

Figure 5.9: Sequence 1 - Search

5.2.2 Personalize Module

Personalization can be performed before a searching by expanding/refining a user query or after a searching by filtering/re-ranking retrieved documents [16]

Pre-processing: query reformulation

Long queries are reduced based on quality predictors and short queries are expanded with additional information [16]. Query expansion refines users' initial queries with additional data that could be relevant to the initial queries so that the system might better satisfy the users' interests. For instance, a researcher might not be familiar with term definitions if he/she is searching a new area of research. A system could consider synonyms of entered search queries or auto-correct to the exact term definitions. Query pre-processing techniques were not incorporated in iScholar at this moment. How iScholar post-processes search results by re-ordering them is discussed in the following section.

Post-processing: re-ranking retrieved results

The returned results from Google Scholar are deterministic. In other words, the same results are returned for the same search term every time. According to Google Scholar[41], a document is weighted and ranked by various factors such as where it was published and who it was written by, as well as how many times it was cited. As referred earlier in Algorithm 3, iScholar adds personalization to these Google Scholar results. The list is adjusted by individuals' publications list and reading history so more similar publications with an individual's personal data are shown in the higher position. Inputs for this module are searched papers, user profile, reading history and the output is the personalized searched paper list.

5.2.3 Recommender Module

This module recommends papers to read, conferences to attend, and conferences/journals to submit.

Recommend papers to read

The factors iScholar considers when it recommends papers are a user's academic interests, past works, downloaded papers, and reading history. Algorithm 4 gives the idea how the system models the factors and generates recommendations. Weight of each factor is approximate based on the importance of each in my opinion, leaving further consideration and experiment.

Algorithm 4 Recommendation - Papers to Read

```
procedure RECOMMEND_PAPERS(candidate_papers, user_profile)
    for each candidate paper do
        build the feature vector (refer Section 4.5.2)
        set $similarities$ to 0
        for each academic interest ($Weight0.35$) do
            $similarity[i]$ += SIM(paper, academic interest)
        end for
        for each past work ($Weight0.35$) do
            $similarity[i]$ += SIM(paper, past work)
        end for
        for each downloaded paper ($Weight0.1$) do
            $similarity[i]$ += SIM(paper, downloaded paper)
        end for
        for each reading history ($Weight0.2$) do
            $similarity[i]$ += SIM(paper, reading history)
        end for
        rank the candidate papers according to the estimated $similarities$
    end for
end procedure
```

Recommend conferences to attend

Inputs for this method are the researcher's academic interests and calls for paper database. By comparing the academic interests and category information of each call for paper, output is the recommendation list (Figure 5.10). Algorithm 5 shows the procedure.

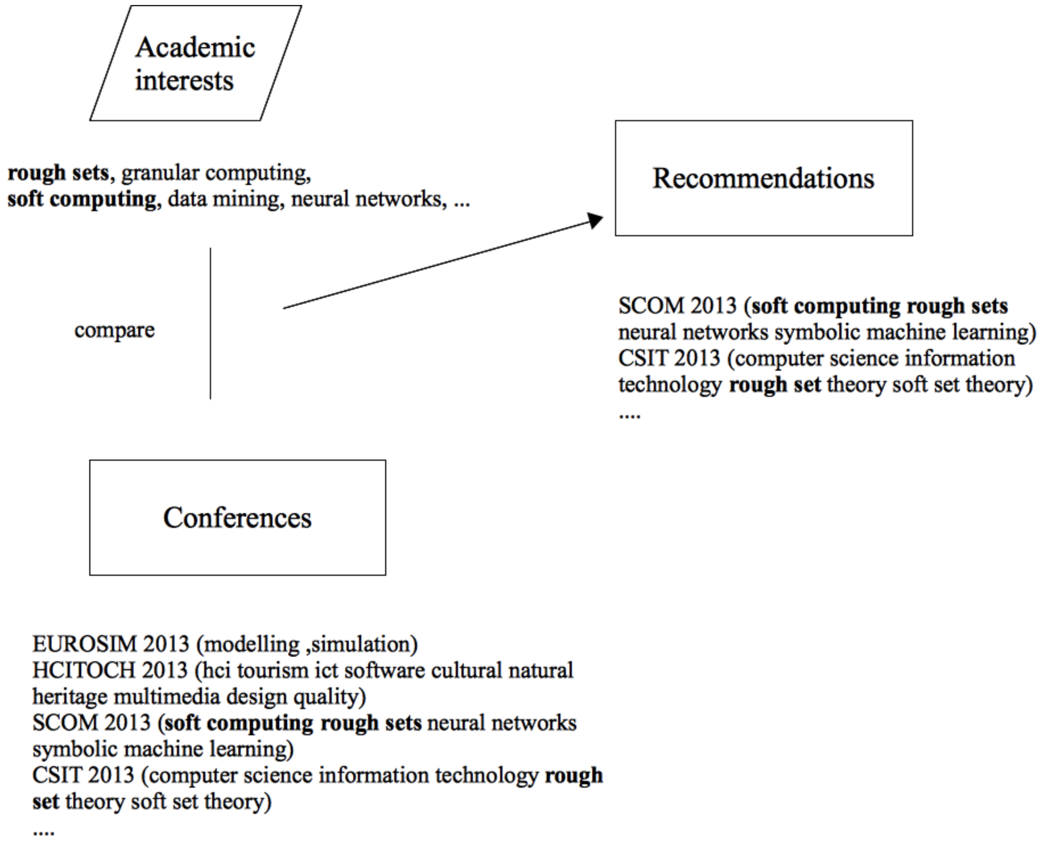

Academic interests
rough sets, granular computing,
soft computing, data mining, neural networks, ...
compare
Conferences
EUROSIM 2013 (modelling ,simulation)
HCITOCH 2013 (hci tourism ict software cultural natural heritage multimedia design quality)
SCOM 2013 (**soft computing rough sets** neural networks symbolic machine learning)
CSIT 2013 (computer science information technology **rough set** theory soft set theory)
....
Recommendations
SCOM 2013 (**soft computing rough sets** neural networks symbolic machine learning)
CSIT 2013 (computer science information technology **rough set** theory soft set theory)
....

Figure 5.10: Recommend conferences to attend

Algorithm 5 Recommendation - Conferences to Attend

```
procedure RECOMMEND_CONFERENCES(interests, conferences)
    empty recommends
    for each conference do
        retrieve categories
        if categories contain interests then
            insert the conference to recommends
            number = how many strings are matched
            insert the number to recommends
        end if
    end for
    sort recommends based on numbers
    return recommends
end procedure
```

Recommend conferences to submit

Conferences to submit are recommended based on a keyword(s), abstract, or entire document. A user starts getting conferences recommendations by entering keywords or an abstract. The system accesses the input and produces recommended conferences to submit. I propose this process as two steps: 1) to extract terms and retrieve candidate papers and 2) to compare the given input with the candidate papers and recommends conferences that the selected papers were published. It is assumed that if two papers are similar then it is likely that the two papers appear in the same conference. So if one wants to publish paper A and paper A/B are similar, it is recommended to submit the paper A to the conference that paper B was published.

1. Whether the given input is an abstract or entire file, extract technical terms, rank them using *tf-idf*, and select top-ranked terms. Then retrieve candidate papers using the selected each term as a searching query.

2. Compare the given input with each candidate paper. Rank again the candidate papers according to the computed similarities and select top-ranked papers. Recommend conferences that each top-ranked paper was published. Display the recommended conferences' information, such as latest calls for paper, research categories, and related calls for papers.

Recommend journals to submit

eTBLAST[42], proposed by Errami et al. [23], is a search engine that finds the best matched journals based on the keywords of the input paragraph of text. The main databases that eTBLAST searches are PubMed[43], PMC[44], and Clinical Trials[45], focusing on medical and biomedical areas. Their idea, searching and recommending journals to publish based on given abstract, can be applied to any academic domain if appropriate databases are utilized. A weighted set of keywords extracted from a query is used to search a database, results are aligned by refining the order and computing scores, and then the Journal Finder suggests the best journals that have published the most similar literature to the query [23].

Lewis et al. [52] suggested the three steps procedure of the text similarity searching using eTBLAST engine: 1) computing vector similarity, 2) measuring weighting score, and 3) ranking by text alignment.

1. Similarity functions: the submitted abstract by a user is converted to a vector representation and the vector is compared with each document in database. For the comparison, three most popular similarity functions - the cosine coefficient, Jaccard coefficient, and Dice coefficient - were introduced.

2. Weighting Schemes: for measuring the importance of each keyword, three difference weighting schemes - tf, td-idf, and slight alteration of the td-idf, were also introduced.

3. Text alignment: compared to the vector alignment, text alignment has an advantage in that it preserves word/sentence order information. Two variations of text alignment - full text and sentence alignment - were examined.

It was concluded that the combination of cosine similarity function weighted with the idf produced the best results [52]. However, the experiment were based on MEDLINE database. It is unknown whether utilizing the other databases will conclude the same result, and the issue remains for further study.

5.2.4 Crawling Module

Compared to journals data, data for conferences needs to be updated more frequently since there is deadline date in conferences. Periodically, crawling module visits the WikiCFP website (refer section 4.5.3) and crawls the updated data for conferences. Listing B.2 shows the source code of this module. The module first opens the webpage, then parses the webpage using Beautiful Soup library, and inserts new data into the conference database (refer Section 4.5.3 - the CFP data representation).

5.2.5 DBManager Module

This module provides the general database tasks such as connecting, inserting, updating, and removing data from database. iScholar uses MySQL for storing conferences/articles data.

5.3 System Evaluation

5.3.1 Accuracy

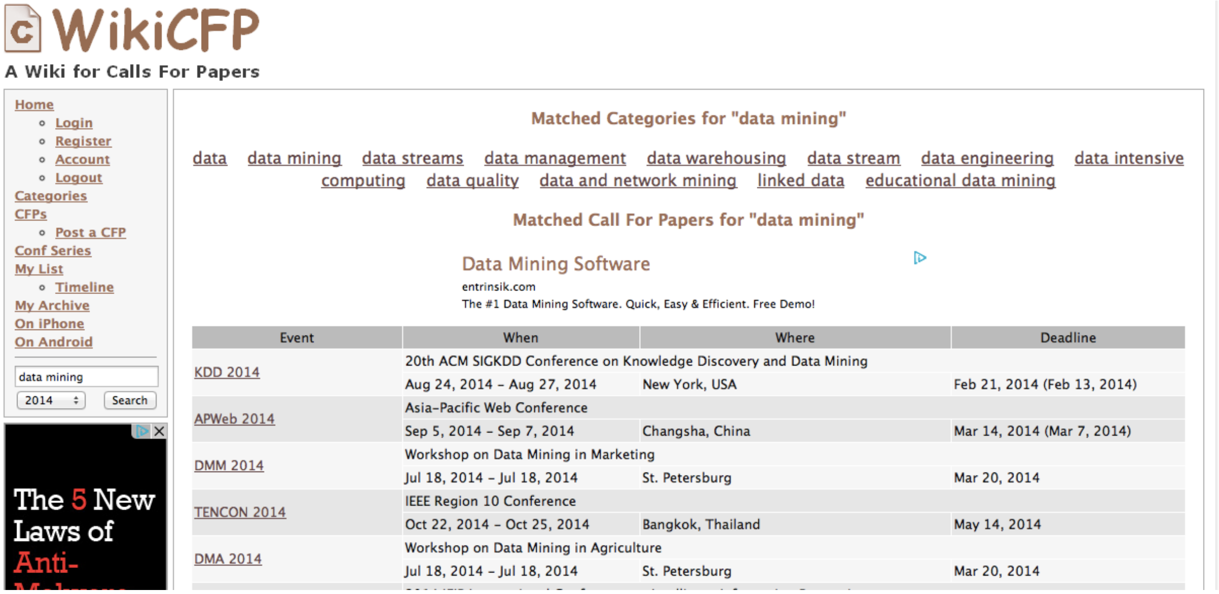
Evaluating a recommender system involves establishing what good recommendations are and how the efficiency is measured [24]. Recommender systems should find the most accurate items to recommend, but there are also other important factors, such as diversity, persistence, serenity, and so on [64]. One advantage of getting recommendations is to have a chance to encounter articles they would otherwise never be found. The recommended articles might come from other disciplines of study or other locations of the world. They bring different viewpoints, which in turn will help to broader one's knowledge. Mobile devices have advantages of getting more personal information as discussed, so chances are higher that mobile recommender systems could generate more relevant and personalized recommendations based on the information.

To evaluate how much accurate recommendations iScholar produces, I performed getting conference recommendations by keyword(s) for different terms and analyzed the relevance between the keyword(s) and recommended conferences (Table 5.5). The time of the experiment was December 2013, and the conferences database contained 177 records.

Table 5.5: Conferences recommendations by keyword(s)

Search term	Recommended conferences
'data mining'	AMT 2014: The 2014 International Conference on Active Media Technology
	BigPMA 2014: The 1st International Workshop on Pattern Mining and Application of Big Data
	BIH 2014: The 2014 International Conference on Brain Informatics and Health
	COPEM 2014: Special Issue of Neurocomputing journal on Solving Complex Machine Learning Problems with Ensemble Methods
	ECMLPKDD 2014: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases
	GeoRich 2014: First International ACM Workshop on Managing and Mining Enriched Geo-Spatial Data 2014
	IDA 2014: The Thirteenth International Symposium on Intelligent Data Analysis
	MSDM 2014: Meeting on Statistics and Data Mining
	PAKDD-SocNet 2014: International Workshop on Data Mining in Social Networks
	WSCBD 2014: Web-scale Classification: Classifying Big Data from the Web
'web intelligence'	WI 2014: International Conference on Web Intelligence
	WIB 2014: Web Intelligence and Bioinformatics

For comparison, I used the search box from WikiCFP with the same search terms (Figure 5.11). In the case of 'data mining', iScholar produced ten recommendations and WikiCFP produced 46 searching results. It is noted that WikiCFP matches both keywords ‘data’ and ‘mining’, whereas iScholar only matches the entire phrase ‘data mining.’ Therefore, WikiCFP produces broader searching results but some conferences of the list seem to be irrelevant to the search term. On the other hand, iScholar only matches conferences that were classified in the data mining category.



Matched Categories for "data mining"

data data mining data streams data management data warehousing data stream data engineering data intensive computing data quality data and network mining linked data educational data mining

Matched Call For Papers for "data mining"

Data Mining Software
entrinsik.com
The #1 Data Mining Software. Quick, Easy & Efficient. Free Demo!

Event	When	Where	Deadline
KDD 2014	20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining		
	Aug 24, 2014 - Aug 27, 2014	New York, USA	Feb 21, 2014 (Feb 13, 2014)
APWeb 2014	Asia-Pacific Web Conference		
	Sep 5, 2014 - Sep 7, 2014	Changsha, China	Mar 14, 2014 (Mar 7, 2014)
DMM 2014	Workshop on Data Mining in Marketing		
	Jul 18, 2014 - Jul 18, 2014	St. Petersburg	Mar 20, 2014
TENCON 2014	IEEE Region 10 Conference		
	Oct 22, 2014 - Oct 25, 2014	Bangkok, Thailand	May 14, 2014
DMA 2014	Workshop on Data Mining in Agriculture		
	Jul 18, 2014 - Jul 18, 2014	St. Petersburg	Mar 20, 2014

Figure 5.11: Conference search - WikiCFP

A decision on usefulness of the ten recommendations may vary from one researcher to the other. As a junior researcher who has not much experience on data mining, I find eight conferences out of ten recommendation look strongly related to the area but the rest two may not be very relevant to my interest (i.e., AMT 2014 and BIH 2014). Nevertheless, the serenity, one of other important factors of recommender systems that were discussed earlier, is showing by recommending multidiscipline areas, for example data mining techniques in the medical area which would have been difficult to find without recommendations.

Table 5.6 shows more search results with different terms. Whether returned conferences are related to the search terms cannot be decided systemically, so I examined the abstract and provided links for calls for papers.

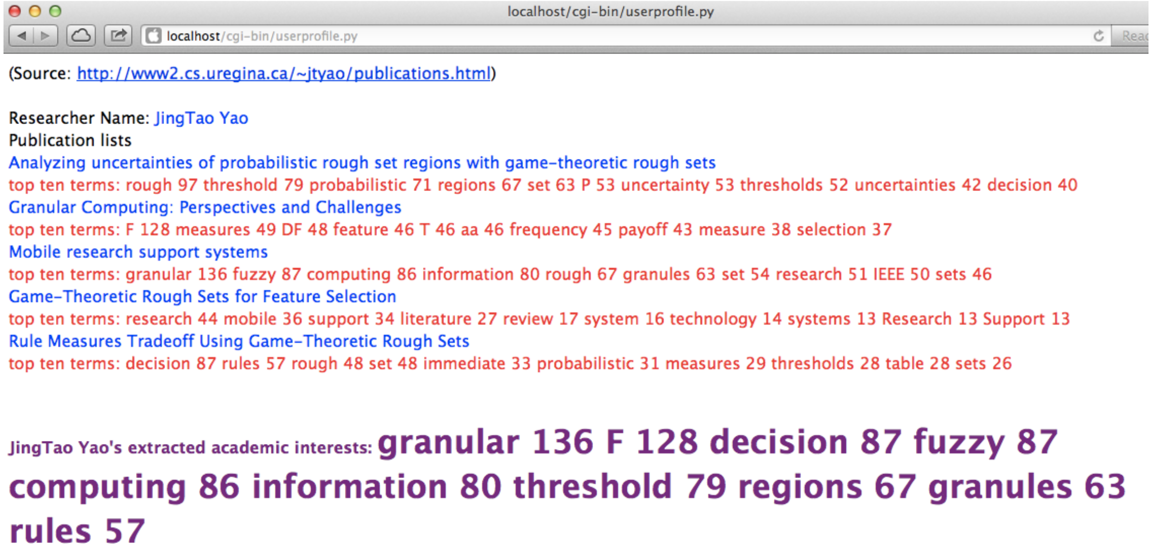
Table 5.6: Conferences recommendations matrix

Search term	Recommen-dations	Related conferences	Irrelevant conferences	Accuracy
artificial intelligence	11	7	4	63.64%
software engineering	8	6	2	75.00%
network	20	16	4	80.00%
cloud computing	7	5	2	71.43%
rough sets	1	1	0	100.00%

It is concluded that in the case of recommending by keyword(s), since no personalization is involved in this case, the quality of database decides the recommendation results. If the title and category of conferences include right keyword(s), the conferences will appear in search results. Not only titles and categories, further considerations on abstract field would be helpful to find out more related conferences. Moreover, contextual and personal information may be also integrated in this case of recommending conferences to submit.

5.3.2 Building User Profiles

To demonstrate how iScholar extracts academic interests from previous publications, I used five recent publications of my supervisor[46]. In figure 5.12, the top ten terms from each paper were extracted first, then among the fifty extracted terms the highest top ten terms were selected to his academic interests. The result show that two selected terms are rather general (i.e., computing and information) and one is not very meaningful (i.e., F), but the other seven actually reflect his academic profile. Only five publications were considered in this case, but if the full list of his publications are used the closer terms would be extracted.


localhost/cgi-bin/userprofile.py
localhost/cgi-bin/userprofile.py
Read

(Source: http://www2.cs.uregina.ca/~jtyao/publications.html)

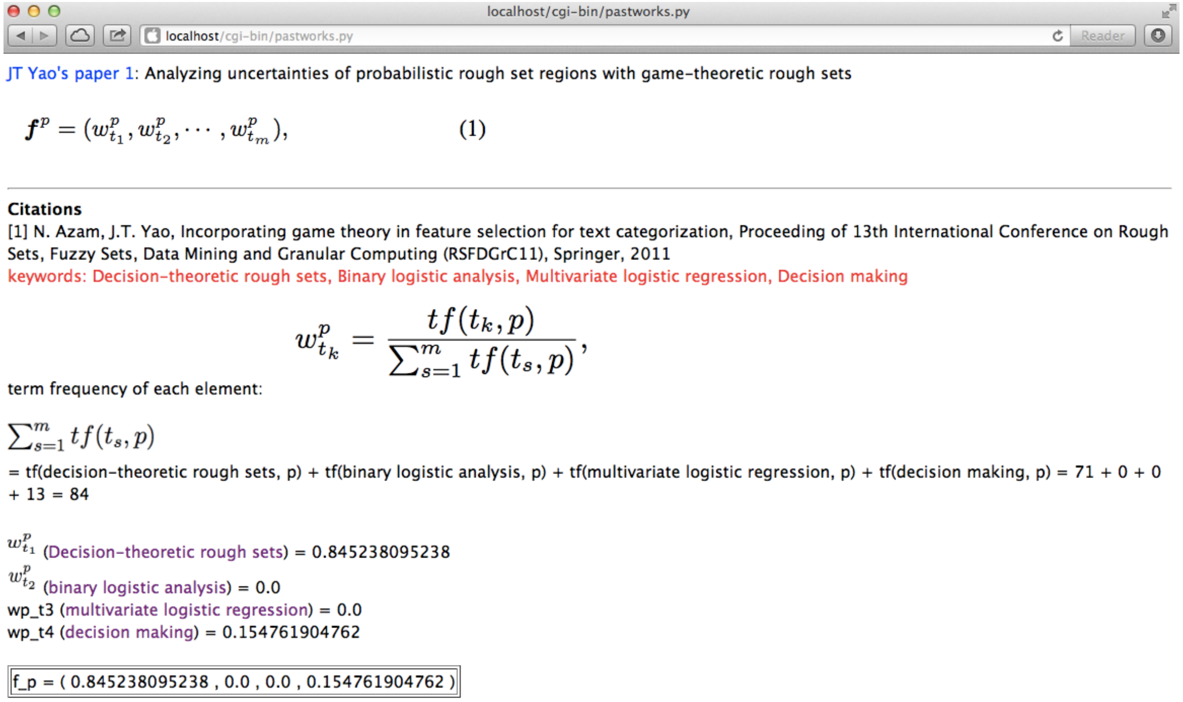
Researcher Name: JingTao Yao
Publication lists
Analyzing uncertainties of probabilistic rough set regions with game-theoretic rough sets
top ten terms: rough 97 threshold 79 probabilistic 71 regions 67 set 63 P 53 uncertainty 53 thresholds 52 uncertainties 42 decision 40
Granular Computing: Perspectives and Challenges
top ten terms: F 128 measures 49 DF 48 feature 46 T 46 aa 46 frequency 45 payoff 43 measure 38 selection 37
Mobile research support systems
top ten terms: granular 136 fuzzy 87 computing 86 information 80 rough 67 granules 63 set 54 research 51 IEEE 50 sets 46
Game-Theoretic Rough Sets for Feature Selection
top ten terms: research 44 mobile 36 support 34 literature 27 review 17 system 16 technology 14 systems 13 Research 13 Support 13
Rule Measures Tradeoff Using Game-Theoretic Rough Sets
top ten terms: decision 87 rules 57 rough 48 set 48 immediate 33 probabilistic 31 measures 29 thresholds 28 table 28 sets 26

JingTao Yao's extracted academic interests: **granular 136 F 128 decision 87 fuzzy 87 computing 86 information 80 threshold 79 regions 67 granules 63 rules 57**

Figure 5.12: user_profile.py

A point to be considered is that some junior researchers might not have enough number of publications to build his academic interests. To compensate this, it would be useful if the system has a way for users to review extracted interests and modify them manually.

Figure 5.13 shows the sequence of constructing one's past works (refer - Algorithm 2). For each candidate papers the feature vector is constructed. Again for my supervisor's paper, f_p is the calculated vector of keywords for his latest paper. The keywords (e.g., Decision-theoretic rough sets or Binary logistic analysis) were entered manually at this moment. It was found that the value for the second and third are zero in the constructed feature vector. In the case of keywords are composed of more than two word, simply getting term frequency may not work properly because the terms may not always used together in the paper. Further consideration on utilizing multi-terms and long keywords is needed for that case.


localhost/cgi-bin/pastworks.py
localhost/cgi-bin/pastworks.py
Reader

JT Yao's paper 1: Analyzing uncertainties of probabilistic rough set regions with game-theoretic rough sets

$$\boldsymbol{f}^p = (w_{t_1}^p, w_{t_2}^p, \cdots, w_{t_m}^p), \qquad (1)$$

Citations
[1] N. Azam, J.T. Yao, Incorporating game theory in feature selection for text categorization, Proceeding of 13th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC11), Springer, 2011
keywords: Decision-theoretic rough sets, Binary logistic analysis, Multivariate logistic regression, Decision making

$$w_{t_k}^p = \frac{tf(t_k, p)}{\sum_{s=1}^m tf(t_s, p)},$$

term frequency of each element:

$\sum_{s=1}^m tf(t_s, p)$

= tf(decision-theoretic rough sets, p) + tf(binary logistic analysis, p) + tf(multivariate logistic regression, p) + tf(decision making, p) = 71 + 0 + 0 + 13 = 84

$w_{t_1}^p$ (Decision-theoretic rough sets) = 0.845238095238
$w_{t_2}^p$ (binary logistic analysis) = 0.0
wp_t3 (multivariate logistic regression) = 0.0
wp_t4 (decision making) = 0.154761904762

f_p = (0.845238095238 , 0.0 , 0.0 , 0.154761904762)

Figure 5.13: past_works.py

5.3.3 Experience Gained from the Prototype Development

The experience gained throughout the development process of iScholar prototype has revealed several interesting aspects including:

- Privacy issues are important in context-aware systems since the level of user profiles and preferences may disclose substantial personal information about a user. Further integrations of privacy preserving models for context-aware systems [11, 45] will be worthwhile.

- Proactivity is one useful aspect of mobile device because mobile users are often less focused [10]. Some introduced proactive systems are Place-Its [26], a location-based reminder, and Newspy[6], a news aggregation alerter. Though iScholar does not consider proactivity at this moment, an example of future feature extensions of an academic assistant would be an automatic displaying of a presenter name, title of the presentation, and other related information when a user enters a conference's presentation room.

- Evaluating a recommender system involves establishing what good recommendations are and how the efficiency is measured. Accuracy, user satisfaction, and satisfaction of the recommendation provider constitute important aspects of evaluating recommender systems [24]. In iScholar's case, since the designed recommendation part has not been implemented in the prototype yet, concrete experimental data or user surveys cannot be provided, leaving them for further considerations.

- One of the biggest questions would be whether iScholar recommends papers/-conferences that make sense to users. To answer this it has to be first observed that users are actually satisfied with the recommendations. It is noted, however, using past publications and academic interests that are from the past publications might not be very useful if one is transitioning research interests. iScholar uses one's academic interests as a 'search term' to retrieve related articles/conferences and discover 'closer' articles among them based on one's other contextual data (e.g., previous publications and reading history). It only considers past line of research but not current interests that have not been published yet. This direction is not catered to by iScholar at present.

- One of main objectives of iScholar was to offer links to a full-text so mobile users can readily download and store into their devices. While Google Scholar provides the links as possible, not all links are supported. If an article is accepted recently there is higher chance that it appears in the search results but the link to a full-text is not available.

Further, one might question whether any highly relevant papers/conferences are not being recommended. A system might fall into a bias by user profiles and miss important articles/conferences in a research area. It might be helpful if iScholar offers, along with a ‘selective’ list, a list of all retrieved results. In addition, providing different kinds of lists, such as influential authors in one’s field or publications from peer reviews only, would alleviate the bias and convince users about the quality of recommendations.

- How research support systems will apply location and time context from mobile devices in a better way still needs to be addressed. For some scholars, for example those who are looking for a PhD study so needing to know where the professors with similar interests are located, where articles were published and by who conferences were organized might be interesting. By utilizing the locational context from mobile devices, a system might offer graphical representations of recommended articles and conferences. Another example of exploring the location-awareness of mobile devices would be a social sharing one’s interests or search queries. For instance, one could view others’ search queries around the person so he/she might get an idea of what others are interested in a specific location, e.g., one’s office or the location when one attends a conference.

Chapter 6

Conclusion

The introduction of mobile technology has recently improved research endeavors including research planning, literature searching, and data gathering. Yet little attention has been paid to an adaptation of mobile technology into research support systems. Some existing mobile application might be useful when conducting scientific research. However, in some ways each has own limitations. The objective of this research is to integrate and introduce the new technology in building more effective research support systems. Architecture of mobile research support systems (MRSS) is proposed along with the design of a research support system specialized for literature review.

It was the main purpose of this study to identify how mobile technology may help researchers and potential future developments. In order to do so, research processes from previous studies were reviewed first and detailed steps of the selected framework were identified. Real-world examples of mobile aids for each step were surveyed. Among the identified stages, the literature review was highlighted since it is believed to be one of the most important activities during the research process. Mobile tools particularly for literature review were also discussed.

iScholar, a mobile exploring/retrieval support system, was presented. It is an information retrieval and recommender system that works on mobile devices for scholarly purpose. Main functionalities of the system - multi-source database discovering, personalized search results, and papers/conferences recommendations - were proposed. The system architecture was established and the details of the system were designed. Finally, details and algorithms for searching database, personalizing searching, and recommending papers/conferences were explained.

6.1 Implications and Contributions

One of the most important responsibilities of science researchers is being up to date on ones' fields of study because knowledge is accumulating and changing every day. There are many kinds of digital resources such as journals, newspapers and many other products and services that are beyond what one can manage and handle. The problem of making efficient use of such resources is a new challenge. The main goal of iScholar is to reduce researchers' overloaded efforts to find information. The system tries to understand researchers' information needs, and provide suitably personalized search results.

iScholar is a mobile client application and a web-server behind that users can search scholarly articles and get recommendations for articles/conferences. The general principles of a new approach to perform personalization on paper/conference recommendations were outlined, relying on ranking and selection techniques which are crucial steps in personalization. In the modeling phase, a variety of alternatives were introduced and rationales for technical choices were described. A researcher model, a combination of his/her past publications as well as academic interests and reading history, was proposed. Suggested personalized recommendations algorithms for articles/conferences based on the researcher model were followed.

Designing and implementing a good support system is not a trivial task. It involves through understanding of the application domain and profound insights on background technologies. This thesis proposed the architecture and design of a mobile support system for scientific research. It is hoped that the issues and designs this study suggests can enlighten future research and further understanding on mobile research support systems domain.

The thesis addressed mainly the mobile support for scholarly research and by doing so the main contributions of the study are as follows:

1. The domain of scientific research and detailed processes of it are explored and formalized. As stated earlier, this process helps to understand the application domain of support systems. In addition, the importance of literature review is highlighted and also the sub-stages of literature review are recognized.

2. Existing mobile tools that might be utilized at each step are surveyed. For potential future developments of research support systems, ways in which mobile technology may help researchers are identified in the formalized research process and the literature review sub-steps as well.

3. One architecture of mobile research support systems was proposed. The architecture may serve for future literature review support systems design. Furthermore, a mobile research support system, *iScholar*, was designed and a prototype was implemented for verifying the possibility of providing better support systems.

4. New features of a research support system, personalized publications recommendations and recommending conferences/journals to submit/attend, were proposed. Algorithms and detailed procedures of filtering scholarly publications,

as well as generating publications/conferences recommendations were explained.

5. How the system exploits distinguishing characteristics of mobile devices and how the system meets considerations of the characteristics were discussed.

6.2 Future Directions

6.2.1 Experiments

Compared to commercial recommender systems, for example music or books, paper recommender systems are relatively new and potentially more challenging due to the complexity of the textual contents. In some domains, such as movies, consensus on which recommendation technique produces more satisfactory recommendations on specific environment has been established in some degree [64]. In general, a recommendation approach is decided based on the existence of available information and relationship between users (or elements). However, paper recommender systems are still underdeveloped as mentioned, so there needs to be further studies on which is the more appropriate recommendation approach.

6.2.2 Further Implementation

Some proposed features are designed but not yet implemented completely. Incorporating personalization when producing recommendations and recommending journals to submit are designed, but need further implementation. In addition, as one of the future enhancements in bookmark module, users can access their bookmark on the Web with computers so that it becomes easy to manage the list. As for another future improvement, citations in MLA, APA, or Chicago formats of the bookmarks could be generated upon users' request.

References

[1] L. G. Ackerson, editor. *Literature Search Strategies for Interdisciplinary Research: A Sourcebook for Scientists and Engineers.* Scarecrow Press, Plymouth, UK, 2007.

[2] A. Allan. *Learning iOS Programming: From Xcode to App Store.* O'Reilly, Sebastopol, CA, 2013.

[3] G. J. Anderson. *Fundamentals of Educational Research.* RoutledgeFalmer, Philadelphia, PA, 1998.

[4] H. Aveyard. *Doing a Literature Review in Health and Social Care: A Practical Guide.* Open Unversity Press, New York, NY, 2010.

[5] N. S. Baron. Do mobile technologies reshape speaking, writing, or reading? *Mobile Media & Communication*, **1**(1):134–140, 2013.

[6] D. Bawden and L. Robinson. The dark side of information: overload, anxiety and other paradoxes and pathologies. *Journal of information science*, **35**(2):180–191, 2009.

[7] J. Beel and B. Gipp. Google scholar's ranking algorithm: The impact of articles' age (an empirical study). In *Proceedings of the International Conference on Information Technology : New Generations*, pages 160–164, 2009.

[8] J. Beel, B. Gipp, S. Langer, and M. Genzmehr. Docear: An academic literature suite for searching, organizing and creating academic literature. In *Proceedings of the International ACM/IEEE Joint Conference on Digital Libraries*, pages 465–466, 2011.

[9] K. E. Berg and R. W. Richard. *Essentials of research methods in health, physical education, exercise science, and recreation.* Lippincott Williams & Wilkins, Philadelphia, PA, 2008.

[10] Reza B'far. *Mobile computing principles: designing and developing mobile applications with UML and XML.* Cambridge University Press, Cambridge, UK, 2005.

[11] P. Bhaskar and S. I. Ahamed. Privacy in pervasive computing and open issues. In *Proceedings of the International Conference on Availability, Reliability and Security.*

[12] L. Blaxter, C. Hughes, and M. Tight. *How to research.* Open University Press, New York, NY, 2010.

[13] C. R. Bomhold. Educational use of smart phone technology: A survey of mobile phone application use by undergraduate university students. *Program: electronic library and information systems*, **47**(4):424–436, 2013.

[14] P. Bonhard and M. A. Sasse. 'Knowing me, knowing you' — using profiles and social networking to improve recommender systems. *BT Technology Journal*, **24**(3):84–98, 2006.

[15] D. N. Boote and P. Beile. Scholars before researchers: On the centrality of the dissertation literature review in research preparation. *Educational Researcher*, **34**(6):3–15, 2005.

[16] M. R. Bouadjenek, H. Hacid, and M. Bouzeghoub. LAICOS: An open source platform for personalized social web search. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1446–1449, 2013.

[17] H. Bukvova. Research as a process: A comparison between different research approaches. *Sprouts: Working Papers on Information Systems*, **9**(29), 2009.

[18] A. Carlos. Research on the go: Mobile tools for conducting research. *The Reference Librarian*, **53**(4):433–440, 2012.

[19] S. Dhar and U. Varshney. Challenges and business models for mobile location-based services and advertising. *Communications of the ACM*, **54**(5):121–128, 2011.

[20] S. E. Efron and R. Ravid. *Action Research in Education: A Practical Guide*. The Guilford Press, New York, NY, 2013.

[21] T. J. Ellis and Y. Levy. Framework of problem-based research: A guide for novice researchers on the development of a research-worthy problem. *Informing Science: The International Journal of an Emerging Transdiscipline*.

[22] T. J. Ellis and Y. Levy. A guide for novice researchers: Design and development research methods. In *Proceedings of the Joint Conference on Informing Science + Information Technology Education*, pages 107–118, 2010.

[23] M. Errami, J. D. Wren, J. M. Hicks, and H. R. Garner. eTBLAST: A web server to identify expert reviewers, appropriate journals and similar publications. *Nucleic Acids Research*, **35**(Suppl 2):W12–W15, 2007.

[24] J. Beel et al. Research paper recommender system evaluation: A quantitative literature survey. preprint (2003).

[25] P. Bhargava et al. Radiology education 2.0 - on the cusp of change: Part 2. ebooks; file sharing and synchronization tools; websites/teaching files; reference management tools and note taking applications. *Academic Radiology*, **20**(3):373 – 381, 2013.

[26] T. Sohn et al. Place-Its: A study of location-based reminders on mobile phones. In M. Beigl, S. Intille, J. Rekimoto, and H. Tokuda, editors, *Ubiquitous Computing*, volume 3660 of *Lecture Notes in Computer Science*, pages 232–250. Springer, 2005.

[27] B. Evans. The ubiquity of mobile devices in universities – usage and expectations. *Serials: The Journal for the Serials Community*, **24**(Suppl 2):S11–S16, 2011.

[28] D. Fallman. Design-oriented human-computer interaction. In *Proceedings of the Conference on Human factors in computing systems*, pages 225–232. ACM, 2003.

[29] A. Fink. *Conducting Research Literature Reviews: From the Internet to Paper*. Sage Publications, Thousand Oaks, CA, 2010.

[30] A. D. Gaikwad and R. R. Dharmi. Personal approach for mobile search: A review. In *Proceedings of the International Conference on Pattern Recognition, Informatics and Mobile Engineering*, pages 221–224, 2013.

[31] A. Göker and J. Davies, editors. *Information Retrieval: Searching in the 21st Century*. John Wiley & Sons, Chichester, UK, 2007.

[32] Google. Our mobile planet. `http://services.google.com/fh/files/misc/omp-2013-us-en.pdf`, 2013. Last Accessed: 2013-11-07.

[33] A. M. Graziano and M. L. Raulin. *Research methods: A process of inquiry.* Longman, New York, NY, 1997.

[34] P. Hagen, T. Robertson, M. Kan, and K. Sadler. Emerging research methods for understanding mobile technology use. In *Proceedings of the Australia Conference on Computer-Human Interaction: Citizens Online: Considerations for Today and the Future*, pages 1–10, 2005.

[35] J. Hahn. *The Best 100 Free Apps for Libraries.* Scarecrow Press, Plymouth, UK, 2013.

[36] M. Hänse, M. Y. Kan, and A. P. Karduck. Kairos: Proactive harvesting of research paper metadata from scientific conference web sites. In G. Chowdhury, C. Koo, and J. Hunter, editors, *The Role of Digital Libraries in a Time of Global Change*, pages 226–235. Springer, 2010.

[37] M. Hearst. *Search user interfaces.* Cambridge University Press, New York, NY, 2009.

[38] J. L. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pages 241–250, 2000.

[39] A. R. Hevner, S. T. March, J. Park, and S. Ram. Design science in information systems research. *MIS quarterly*, **28**(1):75–105, 2004.

[40] O. Hoeber. Web information retrieval support systems: The future of web search. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence - Workshops (International Workshop on Web Information Retrieval Support Systems)*, pages 29–32, 2008.

[41] O. Hoeber and H. Liu. Comparing tag clouds, term histograms, and term lists for enhancing personalized web search. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence - Workshops (International Workshop on Web Information Retrieval Support Systems)*, pages 309–313, 2010.

[42] D. B. Hoggan. Challenges, strategies, and tools for research scientists: using web-based information resources. *Electronic Journal of Academic and Special Librarianship*, **3**(3), 2002.

[43] T. Hong, Y. Wu, J. T. Yao, G. Wang, and Y. Y. Yao. CUPTRSS: A web-based research support system. In *Proceedings of the Workshop on Applications, Products and Services of Web-Based Support Systems*, pages 21–28, 2003.

[44] Apple Inc. Cocoa fundamentals guide. `https://developer.apple.com/legacy/library/documentation/Cocoa/Conceptual/CocoaFundamentals/Introduction/Introduction.html`. Last Accessed: 2013-11-07.

[45] X. Jiang and J. A. Landay. Modeling privacy control in context-aware systems. *Pervasive Computing*, 1(3):59–63, 2002.

[46] H. Kalb, H. Bukvova, and E. Schoop. The digital researcher: Exploring the use of social software in the research process. *Sprouts: Working Papers on Information Systems*, **9**(34), 2009.

[47] F. N. Kerlinger and H. B. Lee. *Foundations of Behavioral Research.* Harcourt College Publishers, Orlando, FL, 2000.

[48] S. Kim and J. T. Yao. Mobile research support systems. In *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, pages 766–770, 2013.

[49] V. Lee, H. Schneider, and R. Schell. *Mobile Applications: Architecture, Design, and Development.* Pearson Education, Upper Saddle River, NJ, 2004.

[50] P. D. Leedy and J. E. Ormrod. *Practical Research: Planning and Design.* Merrill Prentice Hall, Upper Saddle River, NJ, 2005.

[51] Y. Levy and T. J. Ellis. A systems approach to conduct an effective literature review in support of information systems research. *Informing Science: The International Journal of an Emerging Transdiscipline*, **9**:181–212, 2006.

[52] J. Lewis, S. Ossowski, J. Hicks, M. Errami, and H. R. Garner. Text similarity: An alternative way to search MEDLINE. *Bioinformatics*, **22**(18):2298–2304, 2006.

[53] J. K. Lippincott. A mobile future for academic libraries. *Reference Services Review*, **38**(2):205–213, 2010.

[54] P. Ljungstrand. Context awareness and mobile phones. *Personal and Ubiquitous Computing*, **5**(1):58–61, 2001.

[55] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, 2008.

[56] B. Matthias, D. Schahram, and R. Florian. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, **2**(4):263–277, 2007.

[57] D. McSherry. Explanation in recommender systems. *Artificial Intelligence Review*, **24**(2):179–197, 2005.

[58] H. I. Myrhaug and A. Göker. Ambiesense–interactive information channels in the surroundings of the mobile user. **1**(3):230–233, 2003.

[59] J. Nunamaker, M. Chen, and T. Purdin. Systems development in information systems research. *Journal of Management Information System, Winter 1990–91*, **3**:89–106, 1991.

[60] Z. O'Leary. *The Essential Guide to Doing Your Research Project.* Sage Publications, London, UK, 2009.

[61] K. Powell. A lab app for that. *Nature*, **484**(7395):553–555, 2012.

[62] M. Raento, A. Oulasvirta, and N. Eagle. Smartphones an emerging tool for social scientists. *Sociological Methods & Research*, **37**(3):426–454, 2009.

[63] R. Rafter and B. Smyth. Conversational collaborative recommendation – an experimental analysis. *Artificial Intelligence Review*, **24**(3-4):301–318, 2005.

[64] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. Recommender systems handbook. 2010.

[65] I. Salmre. *Writing Mobile Code: Essential Software Engineering for Building Mobile Applications.* Addison-Wesley Professional, New York, NY, 2005.

[66] K. Sugiyama and M. Y. Kan. Scholarly paper recommendation via user's recent research interests. In *Proceedings of the International ACM/IEEE Joint Conference on Digital Libraries*, pages 29–38, 2010.

[67] D. Tennenhouse. Proactive computing. *Communications of the ACM*, **43**(5):43–50, 2000.

[68] T. Wasserman. Software engineering issues for mobile application development. In *Proceedings of the Workshop on Future of Software Engineering Research*, 2010.

[69] W. J. Wilbur and K. Sirotkin. The automatic identification of stop words. *Journal of information science*, **18**(1):45–55, 1992.

[70] P. Winoto, T. Y. Tang, and G. I. McCalla. Contexts in a paper recommendation system with collaborative filtering. *The International Review of Research in Open and Distance Learning*, **13**(5):56–75, 2012.

[71] W. Woerndl, J. Huebner, R. Bader, and D. Gallego-Vico. A model for proactivity in mobile, context-aware recommender systems. In *Proceedings of the ACM conference on Recommender systems*, pages 273–276, 2011.

[72] J. T. Yao. An introduction to web-based support systems. *Journal of Intelligent Systems*, **17**(1–3):267–282, 2008.

[73] J. T. Yao and Y. Y. Yao. Web-based information retrieval support systems: Building research tools for scientists in the new information age. In *Proceedings of the IEEE/WIC International Conference on Web Intelligence*, pages 570–573, 2003.

[74] Y. Y. Yao. Information retrieval support systems. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, pages 1092–1097, 2002.

[75] Y. Y. Yao. A framework for web-based research support systems. In *Proceedings of the International Computer Software and Applications Conference*, pages 601–606, 2003.

[76] E. Yndurain, D. Bernhardt, and C. Campo. Augmenting mobile search engines to leverage context awareness. *Internet Computing, IEEE*, **16**(2):17–25, 2012.

[77] J. Zhuang, T. Mei, S. C. H. Hoi, Y. Q. Xu, and S. Li. When recommendation meets mobile: contextual and personalized recommendation on the go. In

Proceedings of the International Conference on Ubiquitous computing, pages 153–162, 2011.

Notes

[1] http://www.linkedin.com

[2] http://www.citeulike.org

[3] http://scitation.aip.org/iphone

[4] http://www.info.sciverse.com/mobile/downloads

[5] http://code.google.com/p/scholar-droid/

[6] http://www.acquasys.com/Android/Newspy/Default.aspx

[7] http://www.nosleep.net/9-uncategorised/1-idea-sketch

[8] http://en.linoit.com

[9] http://www.uregina.ca/library/mobile/index.php

[10] http://www.ncbi.nlm.nih.gov/m/pubmed/

[11] http://www.libsuccess.org/index.php?title=M-Libraries

[12] http://www.scopus.com/home.url

[13] http://www.thorstensapps.com

[14] http://www.todordimitrov.de/cfp-app/

[15] http://www.wikicfp.com/cfp/

[16] http://www.lifetechnologies.com/ca/en/home/communities-social/apps.html

[17] https://itunes.apple.com/ca/app/od-260/id353740370?mt=8

[18] http://0x0c.com/LabTimer.html

[19] http://notesplusapp.com

[20] http://www.americanlaboratory.com/Scientific-Mobile-Apps/

[21] http://www.macinchem.org/iphone/mobile-apps.php

[22] http://www.adobe.com/ca/products/adobeconnect/feature-details/adobe-connectmobile.html

[23] http://www.easybib.com/

[24] http://www.nuance.com/for-individuals/mobile-applications/dragon-dictation/index.htm

[25] http://www.refworks.com

[26] http://endnote.com

[27] https://www.dropbox.com/

[28] http://thomsonreuters.com/products_services/science/science_products/a-z/current_contents_connect/

[29] http://www.informatik.uni-trier.de/ ley/db/

[30] http://citeseerx.ist.psu.edu/index;jsessionid=8EC5B2D47A776A9DE46EC29B65C96C8C

[31] http://developer.yahoo.com/contentanalysis/

[32] https://developers.google.com/bigquery/

[33] http://scholar.google.ca/intl/en/scholar/about.html

[34] http://www.confsearch.org/confsearch/

[35] http://www.conferencealerts.com/

[36] http://academic.research.microsoft.com/CFP

[37] http://www.wikicfp.com/cfp/

[38] http://www.informatik.uni-trier.de/ ley/db/

[39] http://nutch.apache.org

[40] http://www.crummy.com/software/BeautifulSoup/

[41] http://scholar.google.ca/intl/en/scholar/about.html

[42] http://etest.vbi.vt.edu/etblast3/

[43] http://www.ncbi.nlm.nih.gov/pubmed/

[44] http://www.ncbi.nlm.nih.gov/pmc/

[45] http://www.clinicaltrials.gov

[46] http://www2.cs.uregina.ca/ jtyao/

Appendix A

Technical Documentation

Various computer science technologies, tools, and skills have been employed to develop iScholar prototype. iScholar works on a web-server to communicate and operate it using a mobile device. This enables the server and client codes stay separate so that making changes in either side does not effect the entire system. Each side is consist of:

- Server side: Python, OS X Server, MySQL Server, BeautifulSoup
- Client side: Object-C

Among many possible technical solutions, the above combination was chosen. Some rationales are: 1) more and more people are using Apple devices, especially iPhone, 2) easy accessibility to App Store for distributing the application, and 3) high compatibility with other Apple devices such as iPad, iPod, etc.

A.1 Alternative Solutions

Below A.1.1, A.1.2, and A1.3 list alternative hardware/platform/database choices for mobile support systems.

A.1.1 Mobile Devices

- Smart phones from other manufacturers such as HTC, Motorola, Samsung, etc.
- PDAs (Personal Digital Assistant): PDAs are mobile devices that function as a personal information manager. Most PDAs have the ability to connect to the Internet. PDAs also have electronic visual display and audio capabilities, enabling them to be used as mobile phones and portable media players.
- Tablets: A tablet PC is a tablet-sized computer that has the key features of a full-size personal computer. It is essentially a small laptop computer, equipped with a rotatable touchscreen as an additional input device, and running a standard (or lightly adapted) PC operating system like Windows or Linux.

A.1.2 Mobile Platforms

- Windows Mobile (Microsoft)[*]: Windows Mobile is a platform for mobile devices based on Windows Embedded CE and used in a wide variety of Windows® phones. Visual Studio, Windows Mobile software development kits (SDKs), and developer tool kits (DTKs) make it possible to create software for the Windows Mobile platform in both native code (Visual C^{++}) and managed code (Visual $C^{\#}$, Visual Basic, .NET).
- Android (Google)[**]: Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

[*] http://msdn.microsoft.com/en-us/windowsmobile/

[**] http://developer.android.com/index.html

- BREW MP (Qualcomm)[*]: Brew (Brew MP, Binary Runtime Environment for Wireless) is an application development platform created by Qualcomm, originally for CDMA mobile phones, featuring third party applications such as mobile games.

A.1.3 API Available Academic Databases

- Microsoft Academic Search[**]: Microsoft Academic Search (MAS) is a free academic search engine developed by Microsoft Research. Microsoft Academic Search provides many innovative ways to explore academic publications, authors, conferences, journals, organizations and keywords, connecting millions of scholars, students, librarians, and other users.

- SciVerse (Scopus)[***]: Application developers can create SciVerse applications to improve research productivity. Also researchers can find and influence applications that customize the way they collaborate with the most relevant researchers, search for insights, and manage or analyze the information that accelerates science.

- PLoS (Public Library of Science)[****]: The PLoS Search API gives developers access to rich data that can be flexibly integrated into applications for the web, desktop or mobile devices. It allows PLoS content to be queried using any of the twenty three terms in the PLoS Search. By opening the PLoS content and data through this API we hope to encourage the development of more tools that will improve the way PLoS users discover and interact with our (and their) content.

[*] https://developer.brewmp.com/home

[**] http://academic.research.microsoft.com/About/Help.htm

[***] http://www.info.sciverse.com/sciverse-applications

[****] http://api.plos.org/solr/faq/

A.2 Development Environment

A.2.1 Object-C

iScholar client is built in Object-C language. It is an object oriented language which lies on top of the C. It is primary used in modern computing on Mac OS X as a desktop language and also on iPhone OS. It consists of many frameworks which are containing dynamic shared library and resources to provide services. iScholar mainly utilizes UIKit and network frameworks to communicate with the server.

Model-View-Controller (MVC) design pattern bases its idea on the fact that model (data) and view are independent logical entities. Controller acts as an interface between the two. Without MVC, data and view are implemented as one object which in turn creates problems such as difficult data management from outside the object, creating different user interfaces with same data logic and difficult modification of user interfaces which are tightly coupled with data.

A.2.2 Python

iScholar server side is built in Python language. It is an interpreted, object-oriented, high-level programming language with dynamic semantics. The syntax is simple and easy to learn, emphasizing readability and therefore reduces the cost of program maintenance.

Python supports modules and packages, which improves code modularity and reuse. The Python interpreter and the extensive standard library are available in source or binary form with no charge, and also can be freely distributed.

A.2.3 MySQL

MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use.

A.2.4 External Library - BeautifulSoup

Beautiful Soup is a Python library designed for quick turnaround projects such as screen-scraping. It provides an easy way for navigating a document and extracting what are needed. It parses HTML documents and creates a parse tree that can be used extract specific data from the documents.

References

http://www.python.org

http://www.apple.com/ca/osx/server/

http://scholar.google.com

http://www.crummy.com/software/BeautifulSoup/

http://developer.apple.com/

http://www.mysql.com

http://developer.yahoo.com/search/content/V1/termExtraction.html

http://www.wikicfp.com/cfp/

http://mysql-python.sourceforge.net

Appendix B

Source Codes

B.1 Search Module

Listing B.1: search_gScholar.py

```
import url parse, BeautifulSoup, urllib, urllib2, os, re
#paper search
#url example: http://scholar.google.com/scholar?q=rough set
q = urlparse(os.environ[‘REQUEST_URI’]).query
text = urllib.unquote(q)

#open the url page and get the results html page
url = ‘http://www.google.com/scholar?’ + q
opener = urllib2.build_opener()
opener.addheaders = [(‘User-agent’, ‘Mozilla/5.0’)]
html = opener.open(url)
html_string = html.read()

soup = BeautifulSoup(html_string)
for gs_r in soup(‘div’, {‘class’:‘gs_r’}):
    for gs_rt in gs_r(‘h3’, {‘class’:‘gs_rt’}):  #extract title
        #print gs_rt.string
        for a in gs_rt(‘a’):
            p = re.compile(r‘<.*?>’)
            title = p.sub(‘’, str(a))
        paperList.append(str(title))
    for gs_md_wp in gs_r(‘div’, {‘class’:’gs_md_wp’}): #extract url
        pdf = gs_md_wp.find(‘a’)[‘href’]
        paperList.append(str(pdf))
        ...
print paperList #return the extracted results
```

B.2 Crawling Module

Listing B.2: build_conference_db.py

```
import mysql.connector, BeautifulSoup, urllib2
#get the newest conference page from WikiCFP
url = ‘http://www.wikicfp.com/cfp/allcfp’
opener = urllib2.build_opener()
html = opener.open(url)
pageSource = html.read()
soup = BeautifulSoup(pageSource) #parse the html page source
#identify the table containing the conference data
alltables = soup.findAll(‘table’, {‘cellpadding’:‘3’})

for tr in rows:
  #extract conference names
  event_tds = tr.findAll(‘td’,{‘rowspan’:‘2’})
  for event in event_tds:
    cfp_event = event.text
    #extract event link
    cfp_link = a[‘href’]
```

```
...
#inset new CFP into database
cnx = mysql.connector.connect(user_name, db_name)
query = (‘INSERT INTO cfps (cfp_event, cfp_link) VALUES (%s, %s)’)
    data_query = (cfp_event, cfp_link)
    data_query = (cfp_event, cfp_link)
cursor.execute(query, data_query)
cnx.commit()
```