ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

COLLEGE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Context Aware Semantic Search Engine for Smart Phones

Tewodros Worku Kerie

A THESIS SUBMITTED TO

THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE

31 March, 2015

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

COLLEGE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Context Aware Semantic Search Engine for Smart Phones

Tewodros Worku Kerie

Advisor: Fekade Getahun (Phd)

Signature of the Board of Examiners for Approval:

Name	**Signature**
Fekade Getahun (Phd)	_____
_____	_____
_____	_____

DEDICATED

To my *Father*

Acknowledgment

First of all, I would like to express my deepest gratitude and respect to Dr. Fekade Getahun for his invaluable comments and guidance devoting much of his precious time.

Next I would like to express my sincere thanks to my Family. I always respect and love you.

My special thanks goes to my best friend Tilahun Desalgn for being with me through all those moments of this research work, good and bad. You are not only a best friend, I better declare you brother. Finally I want to extend my gratitude to my friends and Classmates.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Acronyms

ADT:	Android Development Tools
API:	Application Programming Interface
AVD:	Android Virtual Device
DB:	Database
CAUP:	Context Aware User Profile
GPS:	Geographic Positioning System
HMM:	Hidden Markov Model
HTML:	Hyper Text Markup Language
IDE:	Integrated Development Environment
IR:	Information Retrieval
IRI:	International Resource Identifier
IP:	Internet Protocol
JSON:	Javascript Object Notatation
OWL:	Web Ontology Language
NBA:	National Basketball Association
QE:	Query Expansion
RDBMS:	Relational DataBase Management System
RDF:	Resource Description Framework
RDFS:	Resource Description Framework Schema
RGB:	Red-Green-Blue
SPARQL:	Simple Protocol and Resource Description Framework Query Language
SQL:	Structured Query Language
SWRL:	Semantic Web Rule Language
UI:	User Interface
URI:	Uniform Resource Locator
URL:	Uniform Resource Identifier
VLHMM:	Very Large Hidden Markov Model
W3C:	World-Wide Web Consortium
Wi-Fi:	Wireless Fidelity
WWW:	World Wide Web
XML:	eXtensible Markup Language

Abstract

The World Wide Web is an information system of interlinked hypertext documents that are accessed via the Internet. A search engine is a document retrieval system design to find information stored in a computer system, such as on the WWW. With the exponential growth in web content, the answers provided by traditional search engines by query specific keywords to content has resulted in high recall and low precision. Many queries processed on the World Wide Web do not return the desired results because they fail to take into account the context of the query and information about user's situation and preferences. Some search engine now has semantic functionality, can understand data context, but it is limited to only the context of the query.

Context-awareness refers to the capability of a software application to provide services to their users based on the user's current context. As such, the objective of this thesis is to present context aware semantic search engines for smart phones. We present the architecture of context aware semantic search engines on the top of semantic web. Ontology based context modeling is used to represent user contexts. The user's query history and submitted query is also analyzed to identify the concept of the search. Prototype of the system is also developed to demonstrate and test applicability and effectiveness of the proposed approach.

To evaluate the effectiveness of our approach precision measures were conducted on top 15 retrieved documents. The experimental results showed 71.7 % precision which is higher than keyword base search results on the same dataset which is 41.18 % precision.

Keywords Semantic Search; Context aware Search; Search engine; Semantic web.

CHAPTER 1
INTRODUCTION

1.1.Background

The World Wide Web (WWW) allows peoples to share information (data) from the large database repositories globally. Users require information about real object characteristics, available products in a supermarket, a parking close to home, the nearest restaurant, the post office with shorter waiting time, and so on.

A search engine is a document retrieval system design to find information stored in a computer system, such as on the WWW. The search engine allows one to ask for content meeting specific criteria and retrieves a list of items that matches those criteria. Without search engines it is difficult to access information in website, blog, etc. It is almost impossible to look for one by one website for search information in Internet [1].

There are many search engines that come out to help the users on finding their need, but search engines find it increasingly difficult to provide useful results on the Internet. Survey indicates that almost 25% of web searchers are unable to find useful result in the first set of URLs that are returned [2].

Most of the search engines are keyword-based that is,

- Users submit keywords to the search engine
- Spiders build lists of words found on web sites
- Indexer module builds an index
- The search engine searches its search index for the keyword submitted by the user.
- A ranked list of documents is returned to the user.

There are many problems with keyword searching. Some of them are

- Keyword based searches have a tough time distinguishing between words that are spelled the same way, but mean something different
- Unable to return hits on keywords that mean the same but are not actually entered in the query.

- Unable to understand meaning of data, user and device context.

To solve some of the above problems semantic search, an application of the emerging semantic web, is introduced. According Tim Berners-Lee, the inventor of the WWW, semantic web is the web of data with meaning in the sense that the computer program can learn enough about what the data means in order to process it [3].The semantic web is a framework that allows publishing, sharing, and reusing data and knowledge on the web and across applications, enterprise and community boundaries.

Generally semantics search engines

- Understand polysemy and synonymy.
- Understand meaning of the terms.
- Take into account stop words such as a, and, is, on, of, or, the, was, with, by.
- Allows using Boolean operators to refine search results.
- Is designed to try and understand the context that the words are used within the web page to try and match it more accurately to the user's search query.

Semantic search engine searches the semantic web document against a user query for accurate result. It seeks to improve search accuracy by understanding the contextual meaning of search terms. The problem with semantic search engine is it only tries to understand data context.

Context-aware computing is software that examines and reacts to an individual's changing context [4].The use of environmental characteristics such as the user's location, time, identity and activity to inform the computing device so that it may provide information to the user that is relevant to the current context.

Even though semantic search engine tries to solve some of the problems on the traditional keyword base search still the exact user intent while searching for information is not known. Where the user is, his/her health condition, weather condition, his/her agenda, search history etc. should be included. It is reported that 20% of all Google searches and 53% of mobile searches on Bing have local intent [5]. Considering such user and device context will help the search engine to understand what a user is searching for.

In this work context refers to data context (meaning of the term), user context and device context. Context aware search engine will be used to mean a search engine which consider all this contexts.

1.2.Statement of the Problem

Many search engines have limited clues to understand a user's true search intents and thus can only return roughly the same result for the same query. Some search engine now has semantic functionality, can understand data context, but it is limited to only the context of the data. The search engine doesn't guess what information a user would like to see, based on information about the user (such as location, profession, click history, health condition, weather condition…). As a result, websites tend to show only information that agrees with the user's keyword.

For example, a user with mobile device searches for documents using the keyword "*Hospital around*". The traditional keyword based search engines return bunch of documents containing the keyword "Hospital" or "around". The semantic search engines will add relevant results containing keywords medical institution, medical center, health centre, clinic, infirmary, sanatorium, nursing home, etc for the word "Hospital" and nearby, near, approximate....for the word around. But the problem is semantic search engines don't have any information about the user, where the user is at that particular time. For this particular scenario at least the user location is mandatory so that the user will get results which are more vital for him/her.

There are some works done regarding location and we can also take Google under country domains as a location based search engine but the main problem with such works is their limitation to location and IP based location services.

In addition to the location, the users search log is also important. For the above scenario if the users search history consists of the following keywords (cancer, cancer treatment, cancer symptoms.…) and search for the keyword "*Hospital around*" the search engine should select Hospitals around the user which can treat cancer.

To clearly understand the problem statement let us see additional scenario. For this scenario we have two persons located on two different locations, very hot and very cold, and the users search for cosmetics. Here the search result should be different for these two users. Because some cosmetics are not advisable in very hot locations and some are not in very cold

locations. The weather condition of the current user locations shall be additional criteria in the searching process.

So the problem we are going to solve in this work is on this limitation, their limitation to data context. Device and user context will be added to the existing semantic search engines so that the search engines will be intelligent to find and provide what the user is searching for.

1.3.Objective

1.3.1. General Objective

The general objective of this work is to develop context aware semantic search engine for smart phones.

1.3.2. Specific objectives

The specific objectives of this thesis are:

- Assess existing search engines.
- Study about context aware services, capturing, modeling and representing context.
- Study about application development for mobile devices.
- Assessing and selecting different contexts which are relevant for the work.
- Adopt existing approaches in context modeling for search engine design.
- Propose an integrated querying approach - semantic and context searching approach.
- Propose context aware search engine with its different components.
- Develop a prototype of the context aware search engine.
- Evaluation of the prototype.

1.4.Methodology

In order to achieve the general and specific objectives mentioned before, we employed review of literature, model architecture design and prototype developments.

Review of the relevant literature was conducted to have a better and solid understanding of the research domain. A number of published research papers, complete thesis works, books and web sites in the area of context-aware computing, search engines, semantic web and semantic search engine were studied. Existing works related to context aware and semantic

search engines are studied to visualize their importance towards this research and to identify and point direction in providing solution to identified problems.

Using the inputs from conducted review of literatures, a model of proposed architecture is developed. Various components of the model and internal details of each component is also identified.

Prototype of the proposed work is also developed in order to check the usability and applicability of the work. The following major activities were conducted to develop the prototype.

- Specifying the requirements of the prototype.
- Studying tools, technologies and protocols needed for implementing the prototype.
- Developing of the prototype

1.5.Scope and Limitations

This thesis is intended to develop a context aware semantic search engine for smart phones which can be a broad concept. However, the study is delimited focusing around the following aspects.

- Design the proposed architecture.
- Identifying contexts for searching.
- Prototype development and evaluation of the prototype.

The following are limitations of our work.

- The prototype of this work is limited to android based mobile devices.
- The prototype will be implemented on DBPedia datasets. However due to lack of context representation on DBPedia data sets, we forced to limit our context for testing into location and some static contexts.

1.6.Application of Results

The result of this research is context aware semantic search engine for smart phones in which the user can search as ordinary search engines but the search result will be aware of data, device and user context.

The results of the application help users to find appropriate and important information within a short time. The application is generic so that it can be applicable in different domains. The following are some of application areas

- **In medical areas** if a person is under some health condition with this application he/she can search the web without aware of their health condition, the application will consider their condition while searching the web accordingly. For example if user is searching for a specific medicine, name of the medicine as a search query. The search engine consider the name of the medicine, related medicines, nearby pharmacy's (which have the medicine), user's health history to check any contradictory medicines.
- **In education** areas users can search for documents and the results will be aware of their profession or study area and provide relevant results
- **In tourism** if the tourist is going to visit a country and if this information is in his/her agenda for awhile then the search that he/she will take this into consideration.
- **In transport:** suppose the user is searching for "bus" (as a means of transportation) on the Web. For more specific result if there is any travel schedule made on the users agenda, other means of transportations, nearby ticket offices and related context will be used to filter the result. Bus stations for the place, time, price and related information will be provided.

1.7.Thesis Organization

The remaining of the thesis is structured as follows: Chapter 2 starts with review of literature on context aware computing, traditional search engine, semantic web and semantic searching are provided. Review of related works is presented in Chapter 3. Works that have significant relation with this thesis are assessed. In Chapter 4 the design on the proposed context aware semantic search is presented. Each components of the architecture are discussed in details. Chapter 5 presents an implementation and discussion of a prototype system based on the architecture proposed. Finally conclusion and future works presented in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

Context and context-awareness provides computing environments with the ability to usefully adapt the services or information they provide. Information retrieval can benefit from context information to adapt the results to a user's current situation and preferences. In this Chapter, concepts related to our work such as context aware computing, context representation, context modeling, context reasoning, search engine and semantic searching are presented.

2.1 Context Aware Computing

An application that adapts to changing environments are considered as context aware applications and exhibit the following list of features as stated in Day A. [6].

- Presentation of information and services to user;
- Automatic execution of a service for a user; and
- Tagging of context information to support later retrieval

Context-awareness refers to the capability of a software application to provide services to their users based on the user's current context. It has emerged as an important and desirable feature in distributed mobile systems, since it benefits from the changes in the user's context to dynamically tailor services to the user's current situation and needs [7]. Whether a user is at home, at work or with friends, context-awareness will help to tailor services according to the situation the user is.

According to Korkea-aho, almost any information available at the time of an interaction can be considered as context information [8]. Examples include

- Identity e.g. device id, email.
- Spatial information e.g. location, orientation, speed, and acceleration.
- Temporal information - e.g. time of the day, date, and season of the year.
- Environmental information - e.g. temperature, air quality, and light or noise level.
- Social situation - e.g. who you are with, and people that are nearby;

- Resources that are nearby - e.g. accessible devices, and hosts
- Availability of resources - e.g. battery, display, network, and bandwidth
- Physiological measurements - e.g. blood pressure, heart rate, respiration rate, muscle activity, and tone of voice
- Activity - e.g. talking, reading, walking, and running
- Schedules and agendas.

Context-aware systems provide relevant services and information to users by exploiting context. Context related to human factors can be structured into three categories [9]:

- Information about the user (knowledge of habits, emotional state, biophysiological conditions, etc.),
- The user's social environment (co-location of other people, social interaction, group dynamics, etc, and
- The user's tasks (spontaneous activity, engaged tasks, general goals, etc.). the user's tasks (goal-directed activities or the general goals of users).

Context information can be classified as static and dynamic contexts [10]. Static context refers those aspects of a pervasive system that are stable and mostly it is defined. For example, users language preference, device type, date of birth, profession etc. Pervasive system is characterized by continues changes such dynamic information's are called sensed (dynamic) contexts. Such information can be obtained by hardware or software sensors.

Nowadays Smartphone's embedded with a number of sensors which enables the device to provide services based on the context of the user, data gathered from the sensors. For example Samsung Galaxy S5, contains the following sensors [11]:

- **Accelerometer:** measures the acceleration that the handset is experiencing relative to freefall.
- **Light sensor**: measures how bright the ambient light is.
- **Barometer**: Determines the atmospheric pressure of a user's current location and the altitude. Barometer can calculate how many calories are burned according to the atmosphere pressure and altitude.
- **Hall sensor:** Recognizes whether the cover is open or closed.
- **RGB Light sensor:** Measures the intensity of the light and is applied to the Adapt Display, which optimizes screen to surroundings.

- **Gesture sensor:** Recognizes hand movements by detecting infrared rays that are reflected from the user's palm.
- **Gyro sensor:** Detects the mobile phone rotation state based on three axes.
- **Proximity sensor** to detect the presence of nearby objects without any physical contact
- **Heart rate monitor** which enables users to monitor their physical information**.**
- **Fingerprint sensors** to identify the user

When dealing with context information the following characteristics have to be kept in mind [6, 10]:

- **Context-information comes from heterogeneous sources**.
- **Context-information might need to be enriched to be meaningful.** Much of the context information involved in pervasive systems is derived from sensors. There is usually a significant gap between sensor output and the level of information that is useful to applications, and this gap must be bridged by various kinds of processing of context information.
- **Context-information may applicable only in a certain situation.** Every context information may not be important for every applications or situations. Needs proper identification and aggregation.
- **Context information is imperfect**: information may be incorrect if it fails to reflect the true state of the world it models, inconsistent if it contains contradictory information, or incomplete if some aspects of the context are not known.
- **Context-information is continuously subject to change** Due to the dynamic nature of pervasive computing context-information exhibits a range of temporal characteristics.
- **Information delivered by sensors is often uncertain and erroneous**
- **Context information is highly interrelated**. Different types of relationships exist amongst context information.

Context aware system should be able to cope with those facts. In a context-aware system context information is typically combined with other types of (context) information to infer higher level knowledge. According to W. Schwinger et.al the four determining factors in context aware services are [10]:

1. Context scope

2. Context representation
3. Context acquisition mechanism and
4. Context access method

Context Scope: the scope of context refers to type and size of context properties supported by the system together with the ability to extend them to cope with unforeseen requirements. It is characterized by property, extensibility, validity and availability of context.

Context Representation: Symbolizing of the row context data prior to context management process. There are different context representations or modeling approaches varying in accompanying feature. Mark-up-scheme model, object-oriented model, logic based model, object-relational model, and ontology-based model are some of them.

Context Acquisition Mechanism: It refers to the automation of context acquisition and dynamicity in terms of when the context is acquired being either statically considered at system start up or dynamically at run-time.

Context Access Mechanism: There are two generic ways concerning access mechanism, pull and push. Push access mechanism collects the context before it is required by the context services and pull access method collects the contexts information whenever it is needed there

2.2 Context Modeling and Reasoning

Context modeling refers to a formal specification of entities and their relationship. It is sort of data structure that abstracts the sensed entities' context. There are different context modeling approaches with different features. Strang et al. [12] present a survey of six context modeling approaches: Key-value modeling, markup scheme modeling, object oriented modeling, graphical modeling, logic based modeling and ontology based modeling.

Key-Value models are frequently used and represent the most simple data structure for context modeling which employ matching algorithms using key-value pairs for service discovery. Key-value pairs are easy to manage but lack capabilities for sophisticated structuring for enabling efficient context retrieval algorithms.

Markup scheme models use a hierarchical data structure consisting of markup tags with attributes and content which is usually recursively defined by other markup tags. The strengths of markup scheme context modeling approach are partial validation and applicability to existing markup-centric infrastructures of ubiquitous computing environment.

Scheme definition and a set of validation tools exists which enable type checking for complex types and range checking to some degree for numerical values. However, it doesn't satisfy the requirements such as incompleteness and ambiguity, and quality of information while distributed composition is addressed to some level.

Object oriented models use object oriented techniques. It incorporates advantage of object orientation comprising encapsulation, re-usability and inheritance to cover parts of the problems arising from the dynamics of the context in ubiquitous environments. Object oriented context modeling approach is strong regarding the distributed composition requirement. New types of contextual information as well as new or updated instance can be handled in systems in a distributed fashion. Partial validation, and handling incompleteness and ambiguity are possible.

Logic based models use facts, expressions and rules in a high degree formality to define a context model. Formal system is applied to describe the conditions in a set of rules. Logic based context models have distributed composition as their strength, but partial validation is difficult to maintain. This makes the specification of contextual knowledge very error-prone though the level of formality is high. Quality of information, and incompleteness and ambiguity are not addressed while applicability to existing ubiquitous computing environments seems to remain as a major issue.

Ontology based models represent contextual information using description of concepts and relationships with high and formal expressiveness and reasoning techniques. Ontology based context modeling approaches are strong regarding the distributed composition requirement. Incompleteness and ambiguity are also addressed. Partial validation is possible, and a comprehensive set of validation tools do exist.

Compared to other related models, ontology based models provide the most promising assets for context models with regard to the requirements set. It satisfies the requirements better than the other context modeling approaches.

Contextual IR has been identified as one of the important and central challenges in the area on information retrieval [13]. It has been observed that considering only user's current query is not sufficient, as user queries are short and ambiguous [14]. Additionally, different users or even the same user, phrase different information need using the same query or syntactically similar queries [15]. For example the user may search the word Java which is ambiguous as it

has multiple senses: Java coffee, Java Island and Java programming language etc. In the user query, ambiguities can also exists which do not appear in surface. Because of such ambiguities, search engine generally does not understand in what context user is looking for the information. Hence, it returns huge amount of information, in which most of the retrieved pages are irrelevant to the user. Therefore, IR applications should leverage user's context in order to exploit additional information about the user's information need and thus improve effectiveness of the IR result.

According to Google search in the last decade has moved from give me what I said to give me what I want [16]. User expectations from search have rightly increased. In order to understand what the user want and to return results people really want, not just what they said in their query, user's context play vital role.

2.3 Traditional Search Engine

A search engine is a software system designed to search for information on the World Wide Web. The search engines help users to find relevant information from the web. Google[1], Yahoo[2] and Bing[3] are example of traditional search engines.

To use the search engine, a user submit query as a sequence of terms that describe the content he/she is interested, and in return the search engine returns list of potentially relevant web documents.

Generally search engines perform three basic tasks [17]:

1. They search the Internet or select pieces of the Internet based on important words.
2. They keep an index of the words they find, and where they find them.
3. They allow users to look for words or combinations of words found in that index.

[1] www.Google.com
[2] www.Yahoo.com
[3] www.Bing.com

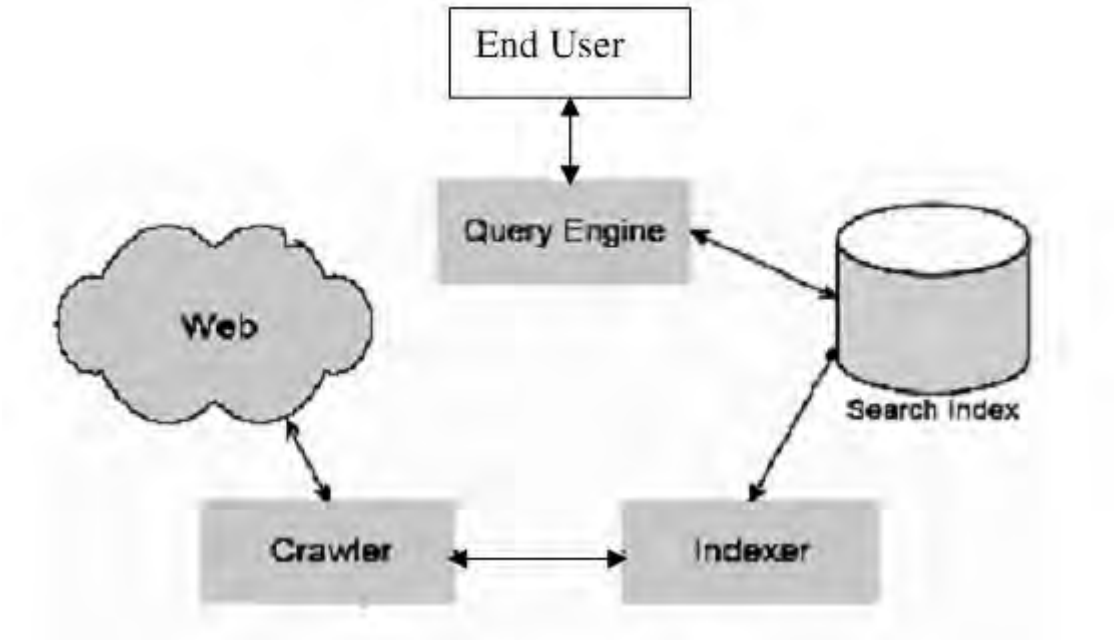


Figure 2-1: Architecture of traditional search engine[4]

A search engine consists of three basic components the crawler, the indexer and the query manager [18].

Crawler (also known as a robot or a spider) is a program that traverses the web to collect web documents and stores them in database**.** It gathers resources (HTML pages, multimedia contents, etc.) from the Web for later analysis by the indexer component. Crawler follows very simple steps [17, 19]:

- It starts with the list of URL's to visit, called seeds and downloads the web page
- It visits the downloaded page and parses through it and finally retrieves, identifies all the links and adds them to list of URLs called the crawl frontier
- For each of the retrieved hyperlinks, repeat the above process.

Indexing is the action of taking the resource gathered by the crawler and building a data structure to be used for searching. An indexer extracts information of document and put in the index file. The purpose of storing an index is to optimize speed and performance in finding relevant documents for a search query.

[4] Picture source http://dbpubs.stanford.edu:8090/pub/2000-37

Query engine is the component that interacts with a user. It accepts the user query, consults the indexer component, makes some ranking, and finally presents the results to the user. The query engine may also perform query expansion in consolidation with a dictionary that contains synonym of words.

Same search on different search engines produces different results. The search result depends on what the crawler finds or what the humans submitted. But more important, not every search engine uses the same algorithm to search through the indexed pages.

The evaluation metrics for search engine, generally IR, are performed by using a document collection, a set of topics describing a user's information needs and a set of relevance judgments, indicating, for each topic, which documents are relevant for each topic. Precision and Recall are most commonly used as performance evaluation metrics. Precision is the fraction of the retrieved documents which are relevant. Recall is the fraction of the relevant documents which has been retrieved

2.4 Limitations of Traditional Search Engine

Current web is the largest globally distributed database that lacks semantic structure and hence it makes difficult for the machine to understand the information provided by the user. The following are some of the key problems of the current web and searching on the current web.

- The web content lacks a proper structure regarding the representation of information
- Ambiguity of information resulting poor interconnection of information.
- Incapability of machines to understand the provided information due to lack of a universal format.

Semantic web and semantic searching are introduced to overcome the above limitation of traditional search engines.

2.5 Semantic Web

Tim Berners-Lee, the inventor of the World Wide Web, defines the Semantic Web as "The Web of data with meaning in the sense that a computer program can learn enough about what the data means [in order] to process it" [3]. Unlike the traditional web which is collection of human interpretable information, semantic web incorporates machine interpretable information.

The semantic web is an extension of the current web that allows computers to manipulate data and information. It allows searching, integration, answering complex queries, content analysis, pattern finding, hypothesis validation, reasoning etc.

The semantic web introduces machine and human processable semantic representation. Allows machines to read, understand and use data over the WWW and accomplish useful goal for user. The semantic web aims to develop a more interconnected Web, where it is easier to find, share and integrate information and services.

The following table describes the difference between traditional web (web 2.0) and the semantic web (web 3.0).

Table 2-1: Difference between traditional web and semantic web

	Web 2.0 Traditional Web	**Web 3.0 Semantic Web**
Main Task	Focus the power of community to create dynamic contents and interaction technology	Linked data, devices and people
Contents	Individual and organization create content	Individual, organization, machine create content which can be reused
Technology	AJAX	RDF
Websites (examples)	YouTube, Facebook, Wikipedia	DBpedia

Building Blocks of Semantic Web

The following figure describes the semantic web framework also referred as the semantic web layer cake by W3C.

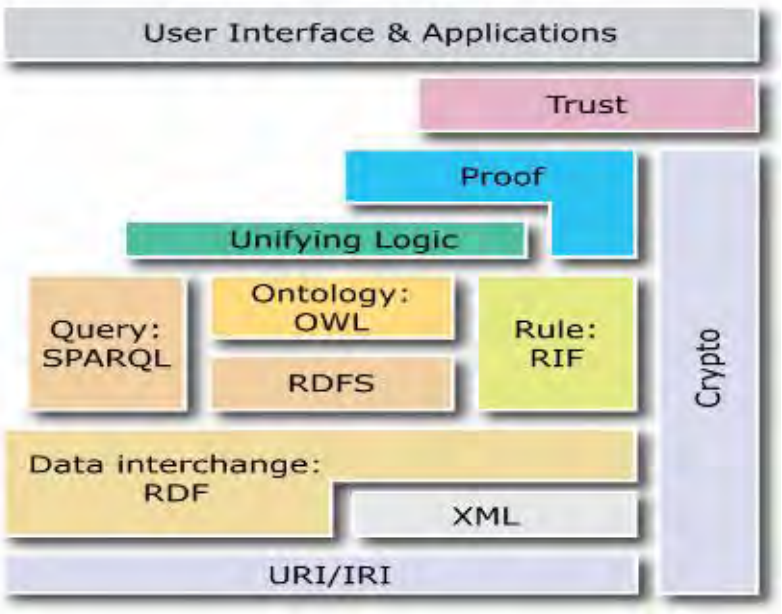


Figure 2-2: Semantic web layer cake[5]

A. Uniform resource identifier (URI)

URI is a unique identifier for a web resource [20]. Such identification enables interaction with resource over a network, typically the World Wide Web, using specific protocols. Uniform Resource identifiers (URI) are used to identify resources on the Web. International Resource identifiers (IRI) extend URIs with Unicode support. A URI does not act as a road map that tells the computer how to get a specific file. Its main function is to identify an Internet resource.

B. XML

XML is a markup language that enables creation of structured data. It provides a surface syntax for structured documents, but imposes no semantic constraints on the meaning of documents. XML namespaces provides a way to use markups from different sources. They are used to refer to different sources in one document. XML Schema is a language for restricting the structure of XML documents. It also extends XML with data types.

C. Resource Description Framework (RDF)

RDF is a general-purpose language for representing information in the Web. It is particularly intended for representing information about Web resources. It is a framework for

[5]Figure source vision for the semantic Web. Available online at: http://www.w3.org/2008/Talks/0307-Tokyo-IH/HTML/img6.html

representing information about resources in a graph form. Information is represented by *subject-predicate-object* expressions. These expressions are known as *triples* in RDF terminology. The subject denotes the resource, and the predicate denotes traits or aspects of the resource and expresses a relationship between the subject and the object.

A directed graph is a bunch of nodes connected by arrows. RDF data are directed, labeled graphs. For example the statement "*The cost of the book 0201000237 is 45.00 Birr*" can be represented as RDF graph as shown below.

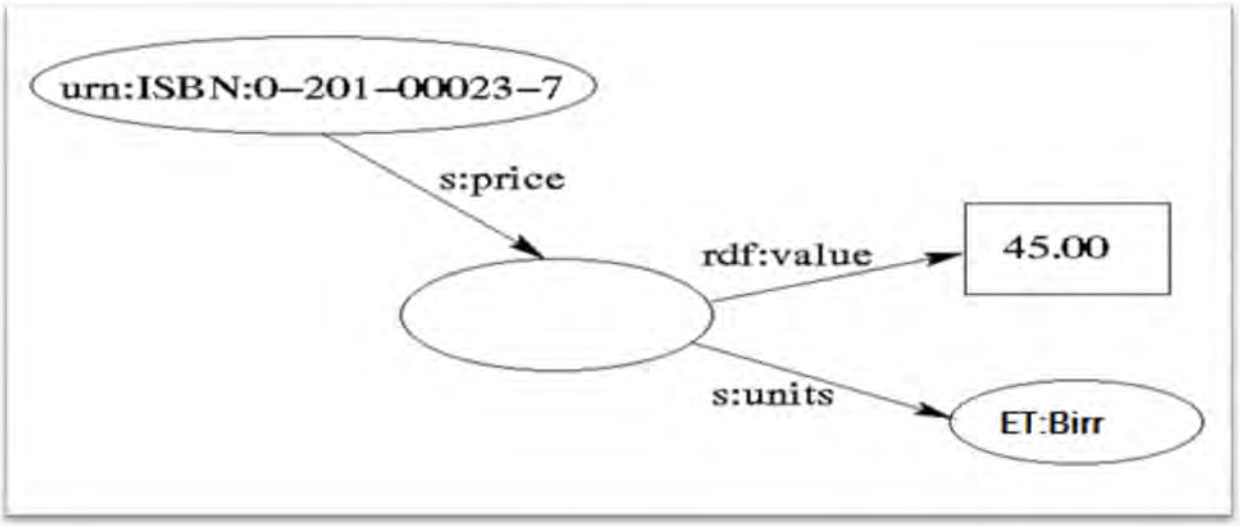


Figure 2-3: RDF graph example

- The node at the beginning of an arrow is called the *subject*.
- The node at the end of an arrow is called the *object*.
- The arrow is called the *predicate*.
- The whole thing is called a *statement*.

There are three kinds of nodes in an RDF directed graph:

- Resource nodes. A resource is anything that can have things said about it. In a visual representation, resources are represented by ovals.
- Literal nodes. The term literal is a word for value.
- Blank nodes. A blank node is a resource without a URI

RDF uses URIs to identify web-based resources (i.e. *subject-predicate-object)* and describes relationships between the resources in terms of named properties and values. However, RDF is only a data model and it does not have any significant semantics. RDF Schema is used to define a vocabulary for use in RDF models.

Several common serialization formats are in use for RDF, including [21]:

- Turtle: a compact, human-friendly format.
- N-Triples: a very simple, easy-to-parse, line-based format that is not as compact as Turtle.
- N-Quads: a superset of N-Triples, for serializing multiple RDF graphs.
- JSON-LD: a JSON-based serialization.
- N3 or Notation 3, a non-standard serialization that is very similar to Turtle, but has some additional features, such as the ability to define inference rules.
- RDF/XML: an XML-based syntax that was the first standard format for serializing RDF.

For example the above RDF graph can be written as follows in RDF/XML format.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#" xmlns:s="http://www.schemas.org/Units/">
<rdf:Description about="urn:ISBN:0-201-00023-7">
<s:price rdf:parseType="Resource">
<rdf:value>45.00</rdf:value>
<s:units rdf:resource="http://www.schemas.org/Units/Birr"/>
</s:price></rdf:Description>
</rdf:RDF>
```

D. Resource Description Framework Schema (RDFS)

RDFS provides basic vocabulary for describing properties and classes of RDF resources [21]. Using RDFS it is possible to create hierarchies of classes and properties. RDF Schema is a language used for describing semantically the classes and the properties of RDF domains. Furthermore, RDFS gives developers the ability to define his/her own RDF ontology, the properties of each object, the relationships between objects and the optimal value which may take each object. RDF and RDFS are a family of World Wide Web Consortium (W3C) specifications [22].

E. Ontology

According to Gruber [23] ontology is a specification of a conceptualization. Conceptualization means structured interpretation of a part of the world in which people used to think and communicate in the world. Ontology provides well-defined meaning for representing a whole domain or part of it to the information enclosed in the web. Ontology describes knowledge about the domain in terms of concepts or vocabularies within the domain and relationships between them.

Ontologies are key requirements for building semantic web for the following reasons [24]:

- A common ontology enables knowledge sharing in an open and dynamic distributed system
- Provide a means for intelligent agents to reason about contextual information,
- Explicitly represented ontology's allow devices and agents not expressly designed to work together to interoperate,
- To enable reuse of domain knowledge.
- To make domain assumption explicit and
- To separate domain knowledge from the operational knowledge.

Depending on the scope of the ontology, ontology may be classified as generic ontology, domain ontology, task ontology and application ontology [25]:

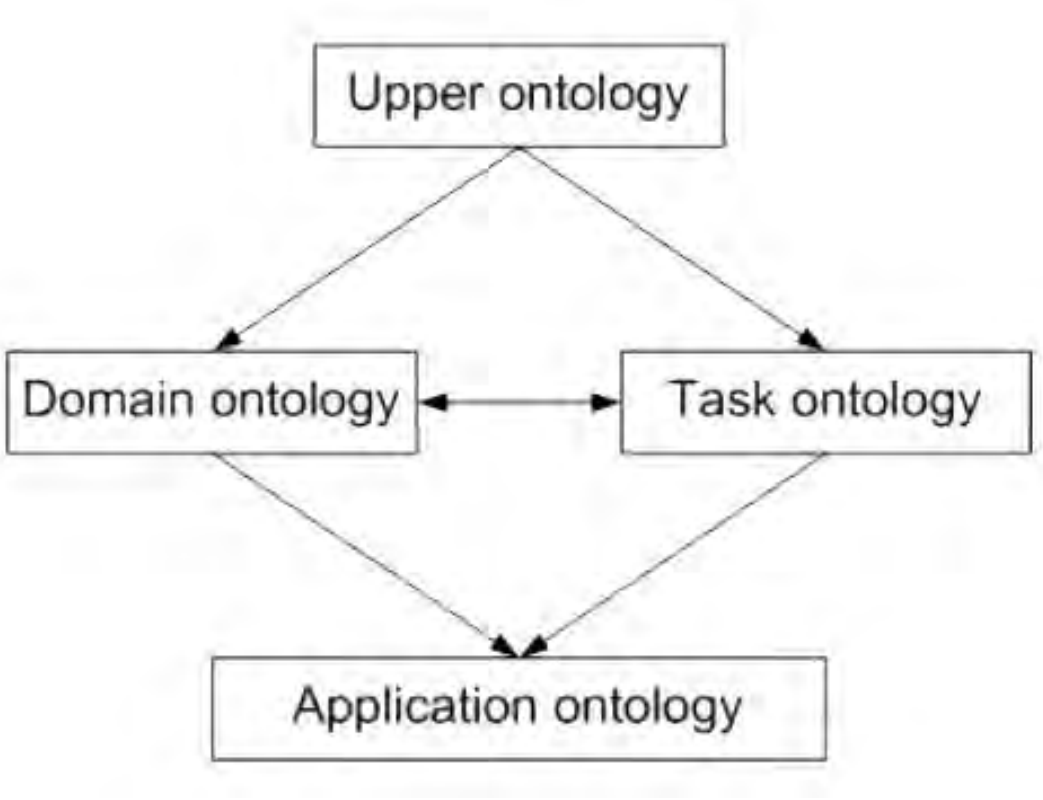


Figure 2-4 : Ontology Classification

- **Generic ontology** - describing general concepts that have a wide scope and are applicable for several domains. Also called, upper, top-level ontologies.
- **Domain ontology** - describing a for specific domain, such as medical domain, engineering domain, or narrower domains, such as personal computers domain
- **Task ontology** –are ontologies suitable for a specific task, such as assembling parts together
- **Application ontology** - ontology developed for a specific application, such as assembling personal computers.

Developing ontology involves the following key activates

- Defining classes in the ontology,
- Arranging the classes in a taxonomic (subclass–superclass) hierarchy,
- Defining slots and describing allowed values for these slots,
- Filling in the values for slots for instances.

F. Web Ontology Language (OWL)

OWL is a language for defining and instantiating web ontologies [26]. OWL ontology may include descriptions of classes, properties and their instances.

OWL is a component of the Semantic Web activity. It became a World Wide Web Consortium (W3C) recommendation in February 2004 [22]. Syntactically OWL ontology is a valid RDF document and as such also a well-formed XML document. This allows OWL to be processed by the wide range of XML and RDF tools already available.

G. Simple Protocol and RDF Query Language (SPARQL)

SPARQL is a query language and data access protocol for the semantic web [27]. SPARQL is defined in terms of the W3C's RDF data model and will work for any data source that can be mapped into RDF. A SPARQL query is composed of:

1. The SELECT clause identifies the variables to appear in the query results and expression that indicates how to construct the answer to the query.
2. The WHERE clause provides the basic graph pattern to match against the data graph

H. RIF

RIF is a rule interchange format. Rules are statements that can be used to infer (discover) new knowledge. A typical kind of ontology for the Web has taxonomy and a set of inference rules. The taxonomy defines classes of objects and relations among them. The inference rules allow an application to make decisions based on the classes supplied without needing to actually understand any of the information provided [3].

I. Cryptography

Is used to ensure and verify that semantic web statements are coming from trusted source. This can be achieved by appropriate digital signature of RDF statements. Trust to derived statements will be supported by verifying that the premises come from trusted source and by relying on formal logic during deriving new information.

J. User interface

UI is the final layer that will enable humans to use semantic web applications.

2.6 Semantic Searching

As with the WWW, the growth of the Semantic Web will be driven by applications that use it. Semantic Search attempts to augment and improve traditional search results by using data from the Semantic Web. Semantic search extends the scope of traditional information retrieval paradigm from document retrieval to entity and knowledge retrieval.

Tim Berners-Lee on, Semantic Web Roadmap, said: "*If an engine of the future combines a reasoning engine with a search engine, it may be able to get the best of both worlds"* [8]. Semantic search improves the results of searches in two ways either by enhancing the list of documents returned by traditional search with relevant data pulled out from Semantic Web or by understanding the sense of the search term to help better filter and sort the list of documents retrieved.

Data publishers state where documents containing RDF data are located, and advertise alternative means to access it. Semantic Web clients and crawlers can use this information to choose the most efficient access method for the task they have to perform.

Generally the goal of semantic search is to deliver the information queried by a user rather than a list of loosely related keyword results. The following are attributes of semantic search (those qualities that make it distinct from non-semantic search) [28].

- Handing morphological variations
- Handling synonyms with correct sense
- Handling generalization
- Handling concept matching
- Handling knowledge matching
- Handling natural language queries and questions
- Ability to point uninterrupted paragraph and the most relevant sentence
- Ability to enter queries freely no special format like quotes or Boolean operators
- Ability to operate without relying on statistical user behavior, and other artificial means
- Ability to detect its own performance

2.6.1 Approaches for Semantic Searching

There are four approaches to semantic search.

1. By contextual analysis to disambiguate queries. The word tank refers to the army tank or storage tank. Such ambiguity will be solved by contextual analysis.
2. Using reasoning, given a set of facts that are represented in the system, additional facts can be inferred from them.
3. Based on natural language understanding.

4. Using ontology that represent knowledge about a domain and expand queries. In this approach, the user query expanded with related concepts. For example the query term "truck," is expanded with "vehicle" as a truck is a kind of vehicle) to make the search more focused as well as more broad).

Most semantic search engines mix and match these approaches in various ways to yield a unique search experience for their users.

Makela describes five mainly used methodologies for semantic search [29]

- RDF Path Traversal - traversing the net formed by a graph of information that uses the RDF data model.
- Keyword to Concept Mapping
- Graph Patterns - used to formulate patterns for locating interesting connecting paths between resources. Also commonly used in data visualization.
- Logics - by using inference based on OWL
- Fuzzy concepts, fuzzy relations, and fuzzy logics

2.6.2 Problems with the current semantic search engines

- Knowledge overhead which is requiring users to be equipped with extensive knowledge of the back-end ontologies and knowledge bases (e.g. form-based search engines) or specific SQL like querying languages (e.g. RDF-based query language fronted search engines) in order to be able to formulate queries or to understand the search result.
- Lack of support for answering complex queries presented by current semantic-based keyword search engines.
- Unable to understand user and device contexts to understand the exact intent of the user.

2.7 Summary

In this chapter we have discussed the main areas that are related to our study. In the first section we have discussed about context aware computing, context modeling and reasoning and characteristics of context. Traditional search engines and limitations of traditional search engines is also discussed in section two.

In the third section we have discussed about the semantic web and related semantic technologies like semantic markup, ontologies, RDF, RDFs etc. In order to understand our study it is important to have knowledge about all these technologies because semantic markup and especially RDF is a crucial point in our study.

Understanding and using user's context information is most important tool in semantic web for improving search results. We introduce context aware semantic search engine to overcome the limitations of semantic search engine. Context concept and its correspondence with Ontologies plays vital rule to make the search results more filtered and personalized.

CHAPTER 3
RELATED WORKS

The ever-growing amount of information available poses significant challenges when it comes to finding specific pieces of information. To tackle this problem a number of works have been developed. In this chapter previous works related to our work will be discussed. We classify related works as works done on context aware information retrieval, location based searching and semantic searching.

3.1 Related Works on Context Aware Retrieval

A number of research works have been conducted on the idea of context aware retrieval. Among these works, we have discussed some of the most related works to our approach.

Gareth J and et al. [30] propose a context aware information retrieval for ubiquitous environment. Context data, user query and user context treated as textual fields. The matching algorithm (retrieval engine) works by computing a score for each field in turn, and then aggregating the results. However matching of textual fields is not an efficient way of retrieval. The matching algorithm may return null result for the specified field but the data may be available with synonym words. The other problem is matching of numeric fields, such as locations given co-ordinates, is a much more open field. To tackle this problem locations are matched using a hierarchical taxonomy. The hierarchical approach can be used to find information like Paris is in France having Paris co-ordinates but it's impossible to find or infer additional knowledge like nearby places.

Context-aware search framework for desktop by taking user activity information into consideration has been proposed [31]. Chen and et al. used Hidden Markov Model (HMM) to capture the relationships between user's access actions (e.g., opening/closing files, sending/receiving emails, etc.) and activity states. The model is learned from users past access history and is used to predict user's current activity upon the submission of some keyword query. They believe the semantic of the keyword queries are related to the context when they are issued. For example, a computer science student John may wish to search for a related paper written by "Michael Jordan" when he is writing his thesis, while he may be

interested in the playing history of "Michael Jordan" when he is reading some NBA news. Architecture of their work is shown on Figure 3-1.

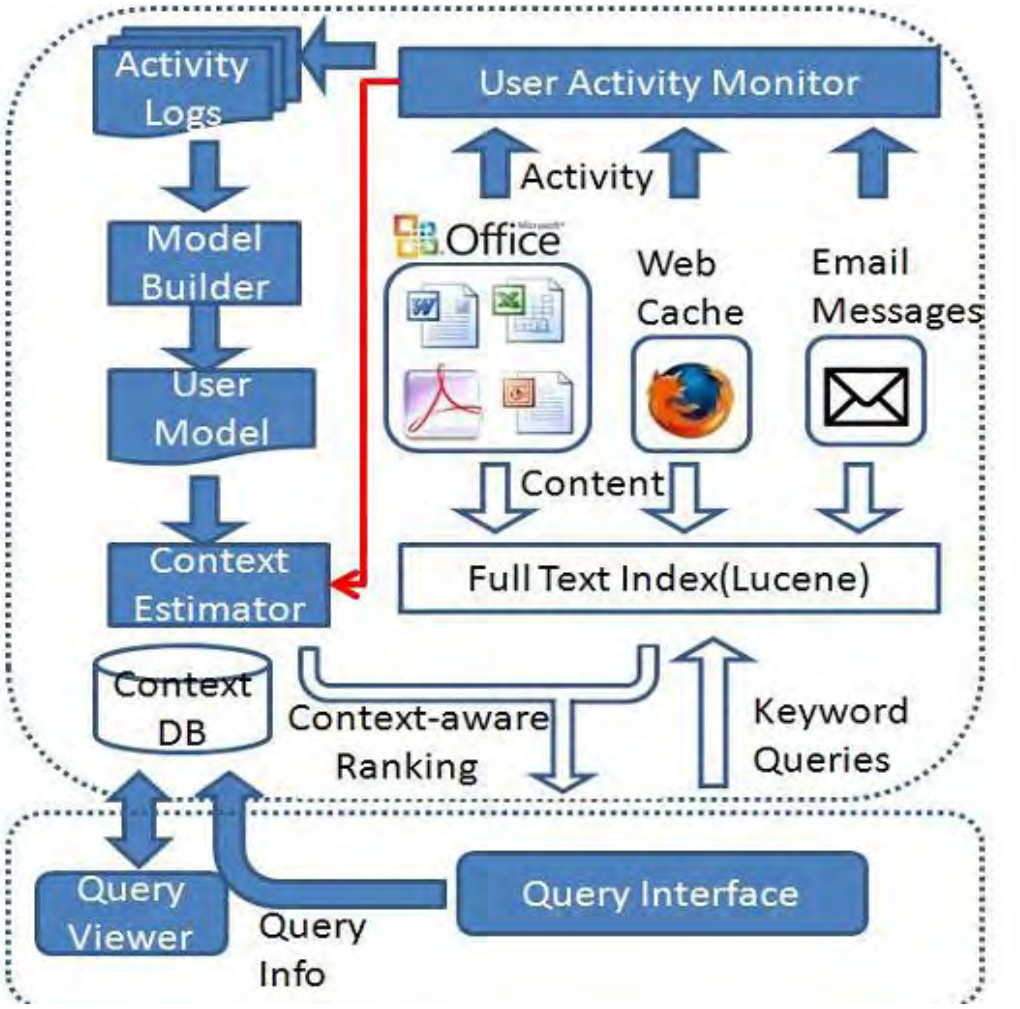


Figure 3-1: Context-aware search framework for desktop

But in the case of web search, such context information (user activity information) usually cannot be automatically captured by web search engines, since due to security consideration, Web browsers are not allowed to monitor user's behavior outside the browser. So it is impossible to apply this approach for web search.

Di Jiang et.al proposed search personalization framework that captures the user's preference in the form of concepts obtained by mining web search contexts. A learning-to-rank approach is employed to combine the derived preference judgments and then a Context-Aware User Profile (CAUP) is created. Huanhuan C. et al propose a general approach to context aware search based on the context of the search query [33]. They Model query contexts by mining search sessions from log data using a variable length Hidden Markov Model (vlHMM).

Huanhuan C. et al develop an algorithms and strategies for learning a very large vlHMM from huge log data and test their approach on a real log data set containing 1.8 billion queries, 2.6 billion clicks, and 840 million search sessions. In [34] session-based search engine using past queries and clickthrough information in the same session is proposed to put the search in context. In this paper the authors also propose an algorithm to recognize the session boundary in order to prevent unrelated queries from interfering in web page ranking. Search session is important mechanism in personalization and context detection. But the drawback of this works is they only captures context of the user from user click-through. Having additional context like user's location, preference and the meaning (semantic) of the search query will help to clearly understand the user's intent while searching.

Chiu C et.al present a search engine built on wireless network in pervasive environments. In pervasive environment sensors can be found on diverse objects such as buildings, cars, and even clothing [35]. The role of the sensors can extend beyond environmental sensing, to become an electronic representation of different objects i.e. description of the underlying objects, or other user defined information. A user can query Snoogle to find a particular mobile object, or a list of objects that fit the description. Snoogle uses information retrieval techniques to index information and process user queries, and Bloom filters to reduce communication overhead. The limitation of this work is searching is limited to physical objects in pervasive environment.

Searching with context Chi Chao Chang et al proposed three algorithms for capturing the information need of a user by automatically augmenting the user query with information extracted from the search context [36]. But in this work the term context is used to represent context term, context of the search query, which is very narrow.

BiaoX and et al presented a context aware ranking method by developing different ranking principles for different types of contexts [37]. Biao and et al adopt a learning-to-rank approach and integrate the ranking principles into a ranking model by encoding the context information as features of the model. The relations between queries in sessions have been used to identify the context of the search.

Daniel A and et.al propose ontology supported personalized search for mobile devices [57]. They propose a context model represent the context-aware information, which will be used to offer a better word recommendation and an auto completion system to improve the user experience. This Context Model uses ontologies and a thesaurus to support a word

recommender system that enhances the typing task and takes into account the user's context and device characteristics.

Shih-Chun Chou and et al present a context-aware museum tour guide that adjusts its recommendations to the interests and contexts of individual visitors and enables them to selectively share their experience with others. The tour guide is built based on semantic web framework to capture and exploit knowledge about the museum's exhibits, its layout as well as contextual information about visitors (e.g. their interests, their locations, how much time they have available, whether they are visiting as part of a group, etc.) and relevant privacy preferences [38]. Context management and context-aware service delivery by Yonas Tadesse [39], Context aware mobile marketing by Anteneh Alemu [40] and Modeling pervasive context-aware museum guide service by Bethelhem Teka [41] are also proposed in our department. But this works are intended to provide service on a certain domain.

3.2 Location Based Search Engines

Jens G and et al presents the GeoSphere search engine for context aware geographic queries on the Web [42]. Geosphersearch facilitates geographic query formation and visual presentation of query results. Geosphersearch first determines the coordinates of the location and expands the query with nearby locations in a default radius using the Geographic Server. If the default radius is not appropriate for the user's needs, it can be adjusted via user feedback ("too far away"), the query is then re-evaluated with an adjusted radius.

Kokono [43] is another work on location based search engines. This system can find web documents based on the distance between locations that are described in web documents and a location specified by a user. Kokono consists of three modules a robot that gathers documents from the Internet, a parser that extracts location information from web documents and converts them into latitude-longitude pairs or polygons and a retrieval module which converts location information specified by user to a latitude-longitude pair and it creates a search circle whose center is this pair. The retrieval module finally it picks up documents that are written about locations within this circle.

GeoSearcher [44] is location-Based Ranking of Search Engine. In GeoSearcher Carolyn W and et al use URL analysis to determine the location and Getty[6] thesaurus of geographic names to map location names onto the geographical coordinates. In GeoSearcher the user

[6]The Getty Thesaurus provides geographic coordinates for places

presents a keyword query and a reference point. The system sends the query and reference point to the search engine and accepts the results. The algorithm then tries to identify the geographic coordinates for the first two hundred sites and calculates the distance from the geographic coordinates for each of these sites to the reference point. These sites are then ranked for the user in ascending order by distance.

This system can retrieve location-related web documents overlooked by conventional keyword-based search engines. Such approaches lead contextualization based on user position. But it is limited to location of the use.

3.3 Related Works on semantic searching

Recently, after the introduction of semantic web, a number of semantic search engines have been proposed and developed. This section briefly discusses some of these works.

TAP [45] presents related information to users using an inference mechanism based on graph traversal. Here, in addition to a traditional keyword search targeted at a document database, the keywords are matched against concept labels in an RDF repository. Matching concepts are then returned in addition to the located documents. The system relies on the Google Web search engine to carry out keyword-based searches and it is limited to the user queries.

TAP Semantic Search, which has two goals:

- Augmenting traditional search results with data pulled from the Semantic Web;
- Using an understanding of the denotation of the search term to improve traditional search, where denotation is the problem of identifying the concepts in the search query;

Main components of TAP infrastructure are

- Information extraction and integration,
- Semantic annotation,
- Semantic data publishing and
- Semantic search

ALVIS[46] open-source semantic-based peer-to-peer search engine, which, before indexing the documents, enriches them with some “semantic awareness” of the specific subject, by using information extraction technology;

Squiggle [47] is a semantic search framework designed to add a conceptual flavor at indexing time and to exploit ontological elements to improve searching time. Squiggle is able to deal with the "meaning" of the searched information based on domain ontology. Squiggle is not a search engine itself, but it allows users to customize their own engine on the basis of particular domain knowledge. The architecture of their work is shown on Figure 3-2.

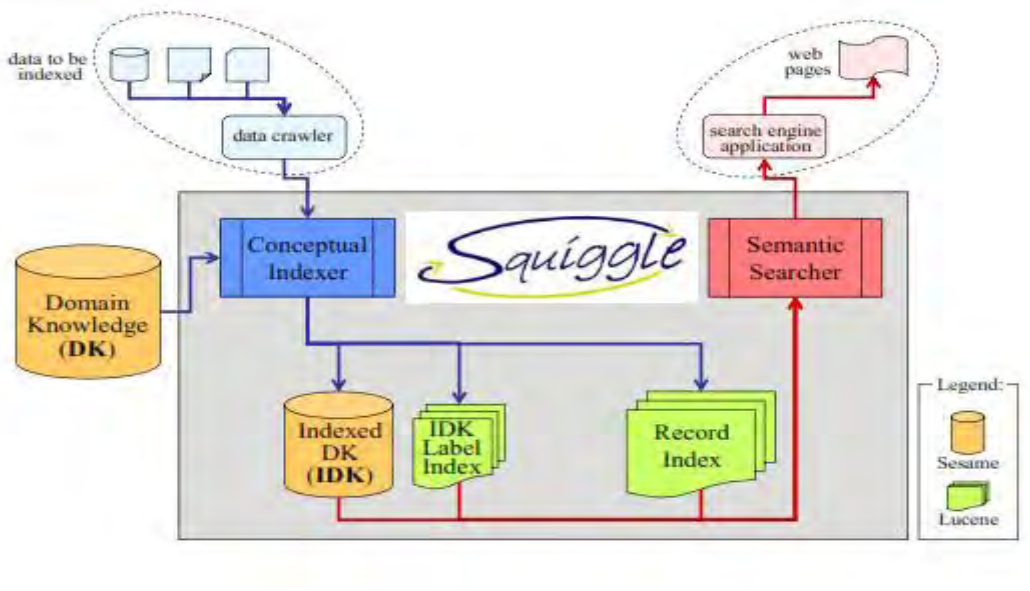


Figure 3-2 Architecture of Squiggle

Salah Sleibi et al present a semantic guided[7] search engine by applying two approaches [48]. The first one is implementing semantic principles through the searching stage, which depends on morphology concept – applying stemming concept – and synonyms dictionary, and the second, implementing guided concept during input the query stage which assist the user to find the suitable and corrected words. They followed the following approach to achieve their goal :

- Improving indexing process by extracting several factors to provide more information about content of the page and retrieve a well and more ranked result.
- Implementing a ranking system which computesthe rank score depending on hyperlink structure and features among web pages and secondly depending on the relevancy between the given query and web pages.
- Improving searching process by applying semantic principles – that means searching not only for the query terms but for their synonyms –

[7]Guided means the suggestion words that appear when the user of search engine typing the query in the search box
using drop down list

- Using guided concept to improve query and help end user when they input query in search box.

Hakia [49] is meaning base search engines around, with a focus on natural language processing methods to try and deliver 'meaningful' search results. Hakia attempts to analyze the concept of a search query, in particular by doing sentence analysis. (*As of April 2014, hakia has terminated its public service, and is now only available for private/enterprise audience.*[8]*)*

Domain dependant semantic search are also introduced [50, 51, and 52]. In [50] uses university domain ontology as a knowledge base for the information retrieval process. The query is analyzed both syntactically and semantically using the domain ontology. [51] Focuses on retrieving information from a sports ontology using the SPARQL query language. The sports related information is queried from the ontology and it is done using SPARQL language

Even though the approach followed by this works (related on on semantic search) is different, they only understand the meaning (concept) of the data and query term.

In [52] relation-centered search functionalities are provided by taking relations between user query terms into consideration. SemSearch provides Google like interface to tackle knowledge overhead [9], support complex queries and generating formal queries based on user queries. The architecture of their work is shown on Figure 3-3

The search process comprises four major steps:

- Query processing to find out the semantic meanings of the keywords,
- Translation of user query into formal query
- Searching the semantic data repositories, and
- Finally ranking the results

[8]https://www.facebook.com/pages/hakiacom/5670874974

[9]requiring users to be equipped with extensive knowledge of the back-end ontologies and knowledge bases or querying languages

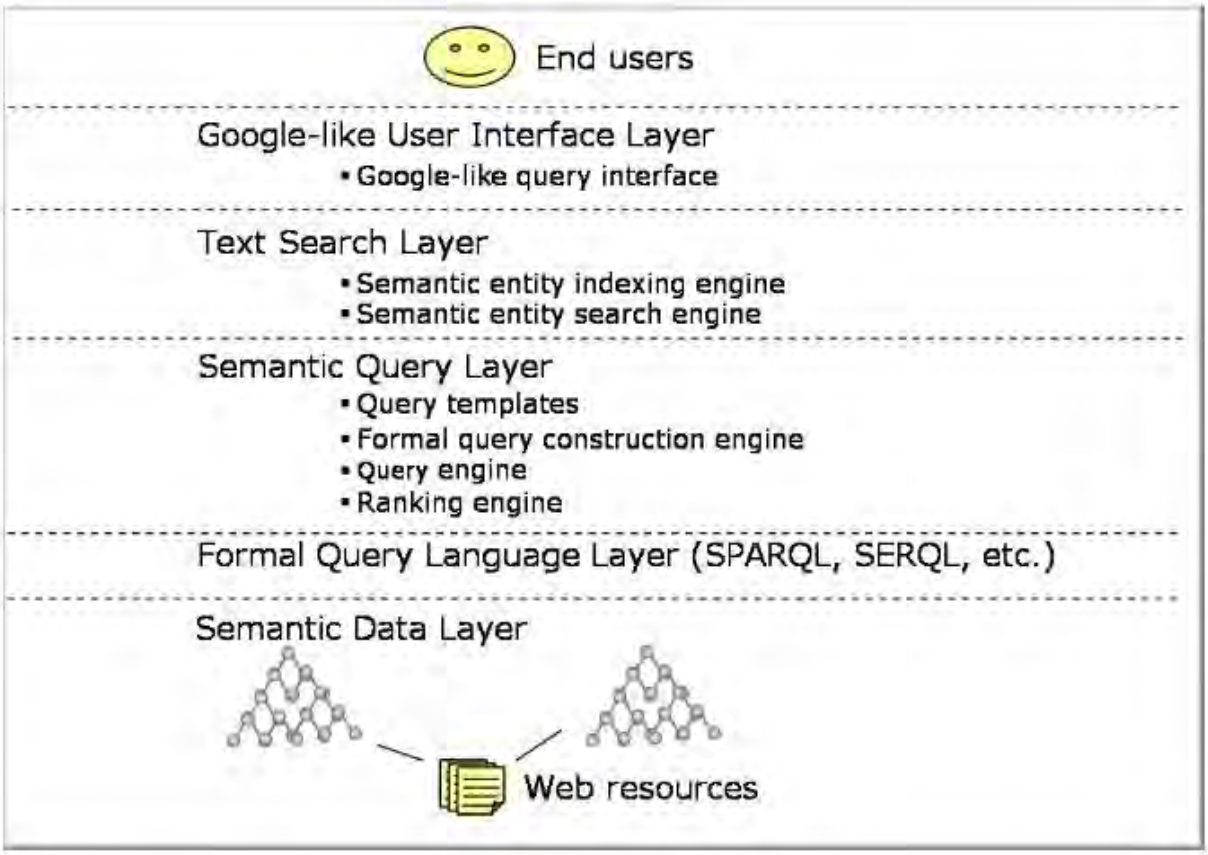


Figure 3-3: Architecture of SemSearch

The formal query construction engine takes as input the matched semantic entities and outputs a set of formal logical query statements based on keywords combinations

The problem with this work is two or more queries may be related or may not be related at all. Having additional context about the user like what he is doing, what he is planning, will solves this problem.

3.4 Summary

In previous works the term context is used either to mean data context, meaning of the search term, or user context such as the user's location, time, identity and activity. Data context is used on different semantic search engines. Semantic search integrates the technologies of Semantic web such as ontology, dictionary and RDF graph traverse and search engine to improve the search results gained by current search engines.

Table 3-1: Summary of related works

	Related Work	Approach	Context Utilize	Limitation
Based on Search Log	Context-Aware Search Personalization with Concept Preference [32]	Through mining click-through and query reformulations	Preference	Limited to preference. Preference is not given explicitly
	vlHMM [33].	Based on the context of the search query by mining search session log data.	Context of search query.	Limited to the context of the search query.
	Session-based search engine [34]	Based on past queries and clickthrough information in the same session	Search log	Limited to the search history of the user
Context aware information retrieval	Cyberguide A mobile context aware tour guide [38]	Based on Semantic Web framework	user's Location and interest	Limited to tourism domain.
	Snoogle [35]	Uses information retrieval techniques to index information and process user queries.	Metadata about the sensors	directly implement the data collection techniques for Internet search engines onto sensor networks
	Context-aware Search for Personal Information Management Systems [31]	search over personal data (desktop search)	User activity	Difficult to trace user's activity for web applications.
	Context aware ranking in web search [37]	Learning to rank approach	The relations between queries in sessions	Integrates context on the ranking stage but having context in before indexing helps to find refined results.

	Ontology supported personalized search for mobile devices [57]	ontologies and a thesaurus use	User and device contexts	Limited to word recommendation and auto completion
Location Based Search Engines	GeoSphere Search [42]	geographic query formation and visual presentation of the result	Location	Limited to location
	Kokono [43]	based on the distance between locations that are described in web documents and a location specified by a user	Location	Limited to location
	GeoSearcher [44]	URL analysis to determine the location and mapping location names onto the geographical coordinates.	Location	Limited to location
Meaning based Search Engines	ALVIS [46],	Based on information extraction technique	Meaning of the search term	Limited to data semantics of results
	Squiggle[47], Hakia[49],	Based on Ontology	Meaning of the search term	Limited to data semantics of query results.

Previous works done on semantic searching enhance the traditional web search. But our main focus is working on their limitation to data context. We propose having user context in addition to the data context for searching improves not only the traditional search but also the results of the current semantic searches.

To the best of our knowledge, there is no previous work on context aware semantic searching which considers the users context together with the user query to search the semantic web in generic domain. Being domain independent or openness makes our work different from the

previous works. So, we take the previous works one step further by integrating the users context together with the user query and by modeling the system in a way that it supports domain independence.

CHAPTER 4
CONTEXT AWARE SEARCH ENGINE

4.1 Overview

In this Chapter, we present the architecture of Context Aware Semantic Search Engine (CASSE) along with its detail components.

Our proposed work is designed on the top of semantic web. As we discussed in chapter two semantic web consist of linked data in the form of RDF and OWL. Searching the semantic web requires RDF triple store dump or accessing through end points. Semantic search engine searches the semantic web document against a user query for accurate result understanding the contextual meaning of terms. In CASSE, we present a novel architecture which integrates user's context in addition to contextual meaning of terms to tailor the search outcome to users.

Figure 4-1 shows a graphical representation of our context aware semantic search engine architecture. CASSE is an adaptation of the classic keyword based search engines described in Section 2.3. However, opposed to the traditional keyword-based search, in CASSE queries are not only gathered from keyword but also from user's preference and context.

CASSE consists of user interface, context manager, query manager, ontology and search log components. The user interface component accepts user query and contexts. Context manager component accept, process, model and reason context information. Query manager is responsible for query expansion and querying the semantic web. Ontology is used for context modeling and query expansion. Our approach also considers the searched history of the user while performing a search. The search log component gathers and stores the search history of the user. The details of the components of CASSE are provided in Section 4.2.

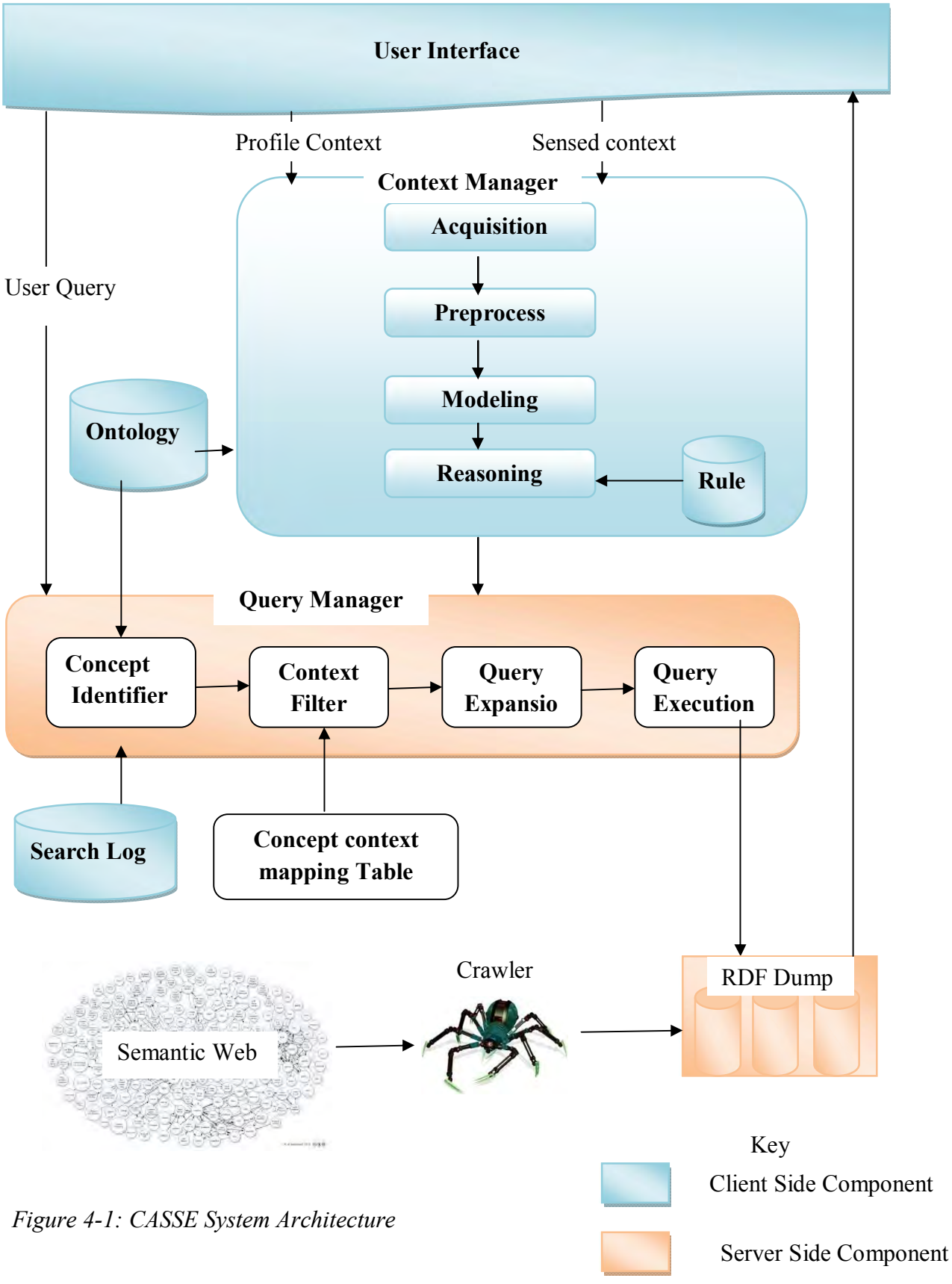


Figure 4-1: CASSE System Architecture

4.2 Major Components

4.2.1 User Interface

The user interface component is client side component which perform the following activities:

- **Provide user interface**: Provide Google like interface for the user to explicitly insert search keyword for searching. Keyword is the query term the user inserts to search the web. This query is similar with keywords used to search for the web in ordinary search engines.
- **Store static contexts:** stores user preference data on local database.
- **Display results:** accepts the results from the server side component and display the search results.

4.2.2 Ontology

Ontology provides well-defined meaning to the information enclosed. The information that is exchanged within a sensor network should have consistent semantics. The semantics of a formal vocabulary can be expressed using ontology.

The proposed context ontology for context modeling consists of domain dependent ontology and the generic domain independent ontology. The generic ontology holds general concepts which are common to the most domains. CASS ontology is used in our CASSE for context modeling and reasoning. Table 4.1 summarizes CASSE generic ontology.

Table 4-1: CASS generic ontology

Class Name	Super Class	Description	Sample Properties
Event	Thing	Used to represent events (music events, historic events, sports events, politics events)	eventLocation, eventTime, eventName.
Location	Thing	Used to model physical locations. defines a set of properties that are common to all concrete physical location classes	latitude locationName locationOwner
Person	Thing	Used to describe people	dateOfBirth dateOfDate name
Device	Thing	Describes device	deviceId manufacturer owner

Class Name	Super Class	Description	Sample Properties
Activity	Thing	Describes the activities in which the user engaged like reading, playing	startTime endTime
Document	Thing	Used to represent documents	documented documentType documentOwner
Organization	Thing	Top class for organization like university, medical institute etc.	organizationName organizationType

Different domain ontologies are designed for different domains. Such domain ontologies will extend from one of the above generic class ontologies. The following table gives sample ontology for Person domain. Classes can have one or many subclasses.

Table 4-2: Sample domain ontology

Class Name	Super Class	Description	Sample Properties
Student	Person	Describes class of people engaged in learning	hasId, hasDepartment, hasName.
Professor	Person	Is a person who teach in a university	hasStudents hasDepartment hasOffice
Doctor	Person	Used to describe medical doctors	hasPatients hasID
Father	Person	Describes peoples who have one or more child	hasChild hasDaughter

4.2.3 Context Manager

The growing number of sensors embedded in mobile devices and the availability of sensor-based applications motivate the use of low level data concerning user's context during searching. Such contexts require a proper acquisition, analysis and management to ensure delivery of the intended services. The context manager component is client side component and responsible for context acquisition, context integration, context modeling and inference.

Our CASSE context manager has context acquisition/sensing, preprocessing, modeling and reasoning subcomponents.

4.2.3.1 Context Acquisition

Sensing component is used for sensing and retrieving raw context data from both physical sensors and virtual sensors (software components). Physical sensor is a device that detects events or changes in quantities and provides a corresponding output, generally as an electrical, digital or optical signal [53]. According to their designed goal the preciseness of the collected data also varies. Virtual sensors are software components for retrieving static contexts about the user.

The software components for context acquisition will be deployed on the users Smartphone to capture contexts. To identify static contexts, interest as well as relevant preferences, the user is asked few questions during the installation time. Such contexts will be stored on a standard database.

Sensing component senses not only the current state of the user but also activates planned for future using user's agenda/calendar. In this work, the following static and dynamic contexts will be gathered from the user, user device.

1. **Location**: geographic location of the user is vital context information for searching the web query concepts. For example, searching for places or items shall consider location of the user at searching time.
2. **Time and week of the day:** such context information is used to get real time search results. For example, searching for restaurant shall consider working time of the restaurant.
3. **Temperature and weather condition:** environmental information is important for weather sensitive results. For example, search result for cosmetics shall be different based on the environment very hot and very cold.
4. **Proximity to other devices & people** such information is used to infer new information about the user. For example, the user's location can be found from who is around him/her or based on devices around the user.
5. **Search log**: Search history of the user consists of search queries, submitting times, clickthrough behaviors etc.
6. **Agenda/ calendar:** future plans of the user. For example, if the user is planning to visit somewhere searching for information should also consider such contexts.

7. **Preference** will be set during the installation and is used to identify user and user preference.

The following are some of static contexts identified for this work.

- Age
- Health condition
- Profession
- Hobbies
- Educational Background
- Living environment (like economic status)
- Working environment

However, these contexts are not equally important in answering a user query. For example, having temperature or weather condition information while searching for 'semantic web' is not important. To solve this problem, we design concept to context mapping. (*See Section* Concept Identification *4.2.4.1*)

4.2.3.2 Context Pre-Processor

The pre-processing of context data aims to make later processing easier by handling missing attributes, cleaning data and mapping to common formats. Different sensors provide information in different formats such data will be mapped in to a common format. The pre-processor component converts this information into a common representation in this case latitude and longitude coordinates. Simple mathematical and statistical metrics can be used to extract basic information from raw sensor data and to perform preprocessing.

4.2.3.3 Context Representation and Modeling

Context information is usually grouped per sub-domains for different intelligent environments such as home domain, office domain, and campus domain. Context in each domain shares common concepts that can be modeled using a general context model, while differs significantly in detailed features.

The context model is based on ontological modeling approach and uses ontology to represent the context. There are several reasons for developing context models based on ontology [7]:

- To define the syntax and semantics of each piece of context information

- Logic Inference to deduce high-level, conceptual context from low-level, raw context, and to check and solve inconsistent context knowledge due to imperfect sensing
- Knowledge sharing to have a common set of concepts about context while interacting with one another.
- Knowledge Reuse. By reusing well-defined web ontology's of different domains (e.g., temporal and spatial ontology), we can compose large-scale context ontology without starting from scratch.

CASS generic ontology discussed on Section 4.2.2 is used to represent domain independent contexts. Figure 5-1 shows domain ontology used to represent contexts for health domain.

4.2.3.4 Context Reasoning

Uses the context model and pre-defined rules to infer new knowledge about a user. Reasoning module used to derive high level context from low level context/s, reconciling context and generate a query parameter and pass the value to the query manager component.

Use the ontology and the rule and context model to infer additional high-level implicit context from low-level explicit context. Table 4-3 shows owl ontology reasoning rules

Table 4-3: OWL ontology reasoning rules

Property	**Rule**
Transitive	(?prdf:type owl:TransitiveProperty) ∧ (?A ?P ?B) ∧ (?B ?P ?C) => (?A ?P ?C)
subClass	(?a rdf:subClassOf ?b) ∧ (?a rdf:subClassOf ?c) =>(?a rdf:subClassOf ?c)
subProperty	(?a rdf:subPropertyOf ?b) ∧ (?b rdf:subPropertyOf ?c) =>(?a rdf:subPropertyOf ?c)
disjoint	(?c owl:disjointWith ?D) ∧ (?x rdf:type ?c) ∧ (?y rdf:type ?d) =>(?x owl:differentFrom ?y)
Inverse	(?p owl:inverseOf ?q) ∧(?x ?p ?y) =>(?y ?q ?x)

A logic based user defined rules used to add additional rules other than owl ontology rules for the inference engine. Applying such rules is a more flexible way of reasoning. Such rules stored in the rule database. Algorithm 4.1 shows the inference (also called reasoning) process.

shown *Figure 4-2* shows an example for logic based rule using semantic web rule language (SWRL).

```
Person(?p), hasAge(?p, ?age), greaterThan(?age, 18)
-> Adult(?p)
Person(?x), hasChild min 1 Person(?x) -> Parent(?x)


parent(?x,?y) ∧ brother(?y,?z) -> uncle(?x,?z)
```

Figure 4-2: Sample SWRL rule

	INPUT
	Context model
	Ontology
	Rule
	OUTPUT
	Inferred query parameter
1.	Load model
2.	Load ontology
3.	Reason on the model
4.	infer new concept
5.	return the inferred model
	END

Algorithm 4-1: Reasoning process.

4.2.4 Query Manager

The main functionality of the query manager is to conduct the query translation and execution. Context manager and user interface provide contexts and user query respectively to the query manager. The query manager translates the application queries to a format understandable by the semantic web retrieval which is SPARQL.

Query manager component consists of concept identifier, context filter, query expansion, and query execution subcomponents.

4.2.4.1 Concept Identification

Concept identifier module used to identify the concept of the search concept based on the search history of the user and concept ontology. According to a research done by Yahoo international search queries can be classified into 24 categories [54]. In our work, we assign contexts for each category. The techniques used to come up with the context of each query categorization is by studying the nature of each category. *Table 4-4* shows search groups and contexts assigned for the groups.

Table 4-4: Search Query Classification

Top Level Concepts (Query category)	Contexts
Health & Pharmacy	Location, weather condition, search log, disease history, economic status
Automotive	Language preference
Consumer Goods	Location, weather condition, Health condition
Entertainment	Hobbies, Location, Activity, schedule
Finance	Language preference, location
Government &Politics	Location
Adult	Age
Art & Humanities	Location
Hobbies	Preference, Search log, Location
International Interest	Hobbies
Life Stages	Language preference
Miscellaneous	Language preference
News	Hobbies , location, schedule
People	Schedule
Reference	Education background, Activity
Religion	Identity, Activity, Location
Trade	Profession
Travel	Preference, Schedule/agenda, Location

Science	Profession
Small Business	Location, Time,
Sports	Team support, sport choice
Technology	Device processor speed screen size, version, availability
Telecommunications	Location, Network Service provider
Uncategorized	

Concept identification module performs semantic matching to identify the concept of the current search. Semantic matching is a technique used in computer science to identify information which is semantically related [55]. Semantic matching between the user previous search history together with the current query is used to align the concept of the current search to one of the above concepts. The concept ontology consists of the above 24 concepts is designed for semantic matching. If the terms (keywords) are related then they are related with some high level concept. The matching algorithm will find this high level concept. Once the concept associated to the search is found, list of contexts for the concept will be returned. Algorithm 4.2 illustrates this phenomenon.

	Input
	Q: string //Search keyword time t qi to qj // set of search keyword within time t
	Output
	search concept C
	Begin
1.	for all q, qi to qj in logdb
2.	Find concept for each query
3.	Check similarity between each concepts
4.	if related
5.	return concept C
6.	Else
7.	return uncategorized
	End

Algorithm 4-2: Concepts Identification

4.2.4.2 Query Expansion (QE)

QE is the process of reformulating original query to improve retrieval performance in information retrieval. Query expansion technology appends related words to query and overcomes word mismatch and improves retrieval. Through query expansion, documents which are on the same meaning can be found even if they do not contain the original query terms. Query expansion includes:

- Semantic analysis to find meaning of the query and to finding synonyms of words. Searching for related terms increases the chances of finding a match with in the index. Our semantic analysis is modeled with RDF graphs. RDF/OWL Representation of WordNet is used to extend the searched terms with three main types of relationships: synonyms, hypernyms and hyponyms.

In addition to the above techniques query expansion component will also integrate context information from the context manager in to the user query. The aggregation is performed using logical conjunction (∧) and disjunction (∨) operators.

For queries constructed using conjunction triple patterns are "and"ed together. The graph pattern is the conjunction of the triple patterns. In each query solution, all the triple patterns must be satisfied with the same binding of variables to values.

For searching additional information, the query should not fail just because some information is missing .For such cases optional graph matching is performed. For each optional block, the query processor attempts to match the query pattern. Failure to match the block does not cause this query solution to be rejected. The graph matching is performed using SPARQL query. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. In an optional match, either the optional graph pattern matches a graph, thereby defining and adding bindings to one or more solutions, or it leaves a solution unchanged without adding any additional bindings. For example the following query finds the names of people in the data. If there is a triple with predicate mbox and the same subject, a solution will contain the object of that triple as well.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE  { ?x foaf:name  ?name .
OPTIONAL { ?x  foaf:mbox  ?mbox }
}
```

4.2.4.3 Query Execution

Query execution module will execute the query against different graphs on RDF dump. A SPARQL query is executed against RDF graph. The RDF graph can be given implicitly by the local API, externally from the SPARQL protocol or it can be specified in the query itself. The FROM clause gives URIs that the query processor can use to supply RDF Graphs for the query execution. A query processor could use this URI to retrieve the document, parse it and use the resulting triples to provide the query graph.

For example, a user searches for hospital using the word "hospital" as a keyword. This query will be translated in to the following SPARQL query.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX type: <http://dbpedia.org/class/yago/>
PREFIX prop: <http://dbpedia.org/property/
PREFIX spatial:<http://jena.apache.org/spatial#>
SELECT ?hospital_name
WHERE {
    ?h a type:hospital ;
    ?h foaf:name ?hospital_name
    ?loc geo:lat  ?lat .
    ?loc geo:long ?lon .
    ?h :hospitalType ?htype.
    ?h spatial:nearby(?lat ?long )
    OPTIONAL { ?h ph:treat 'disease' }
    FILTER (?htype = 'governmental') .
}
```

The query will return the name of the hospitals around the user which can treat a certain disease if the user has previous disease history. The result also selects only certain type of hospitals (either governmental, private based on the economic status of the user.

4.2.5 Semantic Web and RDF Dump

Semantic web introduce semantic representations in to the Web, allowing programs to read, understand and used over the WWW and accomplish useful goals for users. Semantic Web technologies enable people to create data stores on the Web, build vocabularies, and write

rules for handling data. Linked data are empowered by technologies such as RDF, SPARQL and OWL.

A Semantic Web crawler finds the concepts in the annotated Web pages and builds a knowledge base. Sitemap extensions allow data publishers to state where documents containing RDF data are located, and to advertise alternative means to access it, such as data dumps and SPARQL endpoints. Semantic Web clients and crawlers can use this information to choose the most efficient access method for the task they have to perform

The RDF data model expresses information as graphs consisting of triples with subject, predicate and object. Many RDF data stores hold multiple RDF graphs and record information about each graph, allowing an application to make queries that involve information from more than one graph

RDF dump set is collection of large RDF triplets which has been downloaded by the semantic web crawler component. The Triple Patterns are written as a whitespace-separated list of a *subject*, *predicate* and *object.* Query manager execute queries against this RDF triplets.

In our model the semantic crawler crawl RDF data from the Web and stores the crawled data as RDF datasets. A dataset is a set of RDF triples that are managed, stored, or published together. The RDF datasets will be stored on the server side as RDF dump.

CHAPTER 5
IMPLEMENTATION AND DISCUSSION

The previous chapter presented the detail of the proposed context-aware semantic search engine architecture. This chapter discusses the prototype implementation of the architecture by presenting the objectives of the prototype, tools and technologies used, implementation details, demonstration and evaluation of the proposed system.

5.1 Overview

The prototype developed in this study is principally aimed at providing a proof of the concepts of the proposed context aware semantic search. We present use case scenarios in health domain to assess the usability and applicability of the proposed generic architecture.

Usecase Scenario

Alice installed CASSE client application on her android mobile phone and searches for hospitals around her using the keyword "*Hospital*" at 8 am. Previously, Alice searched the web using *cancer*, *cancer symptoms*, and *headache* keywords. She is at (*9.002683, 38.779346)* location while searching the web. Alice's default language choice is *English* language. The economic status of Alice is *Low income* as it is indicated on her preference.

5.2 Tools and Technologies Used

Several tools and technologies are utilized for the purpose of developing the prototype implementation. The following is list of programming, communication, database management, context representation, context reasoning and operating environment used in the prototype implementation.

- Java[10] programming language is used to write the implementation of the prototype. The Java™ Programming Language is a general-purpose, concurrent, strongly typed, class-based object-oriented language. It is normally compiled to the bytecode instruction set and binary format defined in the Java Virtual Machine Specification. It

[10] https://www.java.com/en/

is designed to be platform independent with additional key principles such as usability, reliability and security.

- Protégé[11] is used to construct the ontologies required in the prototype. It is a free, open source platform that provides a suite of tools to construct domain models and knowledge-based applications with ontologies. It has ontology editing environment with support for the OWL 2 Web Ontology Language, and direct in-memory connections to description logic reasoner
- Jena[12] is used to access and manipulate ontology concepts modeled through java programming environment. Jena is a Java framework for building semantic web applications. It provides programmatic environments for RDF, RDFS, OWL, SPARQL and includes a rule-based inference engine. The Jena framework includes a RDF API, an OWL API, in-memory and persistent storage and SPARQL query engine.
- JENA's ARQ module provides full support for SPARQL. It also provides significant extension beyond the SPARQL specification. These includes
    - Regular expression over paths in RDF graph
    - Querying remote SPARQL endpoints
    - SQL like group by, having and aggregation clauses
    - Nested Select support
    - Variable assignment – values of variable can be computed by an expression from other variable values
    - Transitive reasoner, which handles subClassOf and subPropertyOf
- Android Development Tools (ADT)[13] is a plugin for the Eclipse IDE that extends the capabilities of Eclipse is used to set Android projects, create an application UI, add packages based on the Android Framework API, debug applications using the Android SDK tools, and even export signed .apk files in order to distribute application Prototype.
- SQLite is a open source SQL database that stores data to a text file on a device. Android comes in with built in SQLite database implementation. SQLite supports all the relational database features.

[11] Protégé - http://protege.stanford.edu/
[12] http://jena.apache.org
[13] *developer.**android**.com/**tools**/help/adt.html*

- Virtuoso Universal Server is a middleware and database engine hybrid that combines the functionality of a traditional RDBMS, virtual database, RDF, XML, free-text, web application server and file server functionality in a single system. Virtuoso is used as RDF triplesstores and querying RDF triplets using SPARQL.
- DBPedia public SPARQL Endpoint[14] is used to query DBpedia data set.
- The DBpedia Ontology is a shallow, cross-domain ontology, which has been manually created based on the most commonly used infoboxes within Wikipedia.The English version of the DBpedia knowledge base describes 4.58 million things, out of which 4.22 million are classified in a consistent ontology. Table 2-2 below lists the number of instances for several classes within the ontology:

Table 5-1: Instance classes of DBPedia ontology

Class	Instances
Resource (overall)	4,233,000
Place	735,000
Person	1,450,000
Work	411,000
Species	251,000
Organization	241,000

5.3 Setting up the Prototype

To support scenario in the *Section 0* a prototype is implemented as a client-side Android application (Hawaii U8180 or AVD android emulator) and the server-side on context aware search server. The development environment for the client side application is based on an android programming language with a targeted SDK API level 19 (KITKAT) and minimum SDK with API level 8 (Froyo). The server side module is implemented on desktop computer with Windows 7 operating system.

5.4 Implementation Details

In the previous section, *Section 5.3*, we have seen the general prototype setup we have used to test our application both on the client side and the server side. In this section implementations detail of each components is presented.

[14] *http://dbpedia.org/sparql*

5.4.1 Client side application

Table 5-2 illustrates the implementation detail of the components found in the client side.

Table 5-2: Client side module implementation detail

Component name	Implementation detail
Query collector	Implemented as android activity class • Provide interface for the user query • Accept query from the user as a string and provide form to set preferences
Context sensor	Implemented as an android activity class. This module accepts the concept of the search from the concept identifier module and • Read the location of the phone either from GPS location or GSM signal. • Scan for preference for default location if there is a problem in reading the user location and • Reads weather information (Temperature, Humidity) based on the location information.
Context data storage	Implemented using SQLite database in order to persist user's preference (static context information) and any relevant user context.
Search log	Implemented using SQLite database to store search keywords, number of hits and submitted date.
Display	Implemented as android activity class running in the main UI and used to view search results received from the server to the user

5.4.2 Server side Module

The server side module is implemented using java programming language and virtuoso server to query DBPedia datasets. *Table 5-3* illustrates the implementation detail for the modules that reside on the server.

Table 5-3: Server side module implementation detail

Component name	Implementation detail
Concept identifier	• Implemented as java class to read the concept ontology and list of keyword (previously used for searching) within time t. • Returns the concept, if the keywords are related, to the filter module
Context Filter	• Implemented as a java class • Responsible for filtering context data • Sends list of contexts to be gathered for the context sensor module based on the concepts identified.
Context processor	• Implemented as a java class which accept context from context sensor client module preprocess, model and reason the context information. • Uses ontology shown below on figure 5.1 for modeling ad reasoning. • Jena, API for java,is used to creating, reading and writing RDF graphs. • The coordinates converted into country code by the context pre processor component for some to find the of the country.
Context modeling	• Implemented as a java class and uses ontology to represent context *Figure 5-1* shows ontology used for modeling context information.
Context Inference	• Implemented as a java class and uses ontology model and pre-defined rules to infer new knowledge about a user. • For example from Alices's economic income information the inference engine infers new knowledge "*Alice Hospital choice is community hospital"*.
Query Execution	The query execution module is implemented on DBPedia end point. SPARQL query is used to query the datasets.

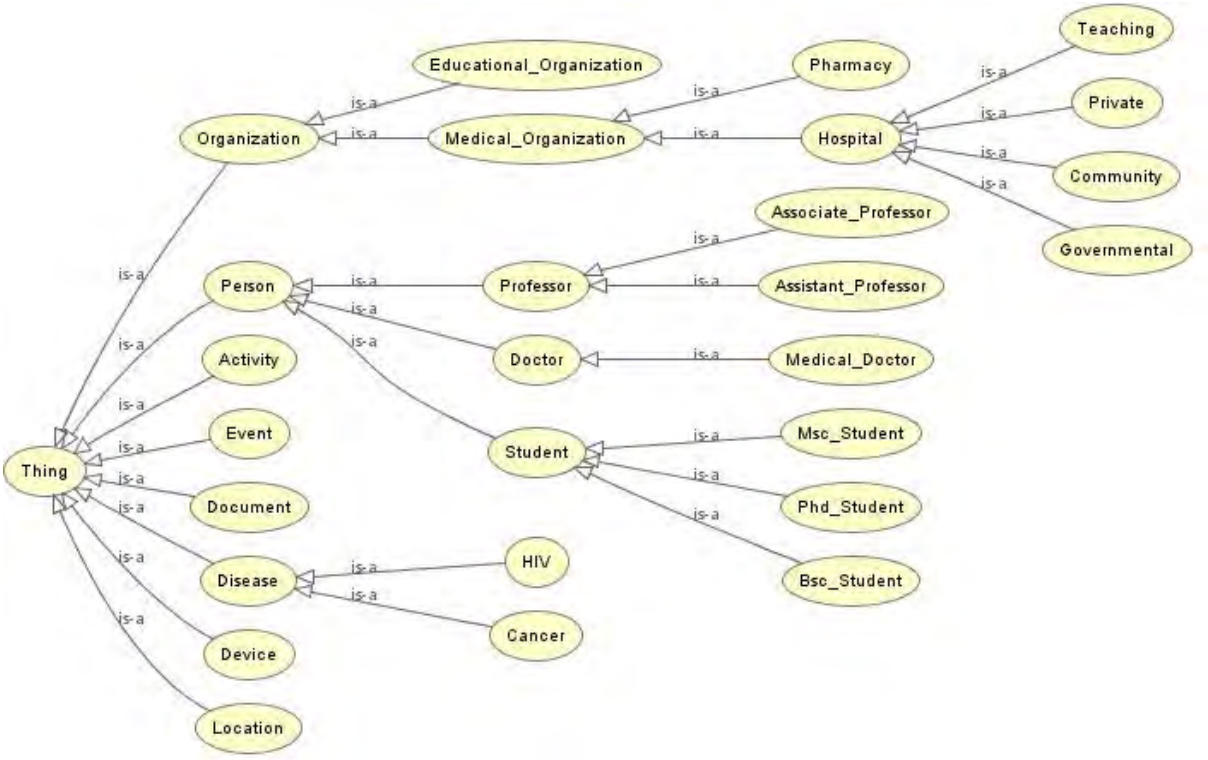
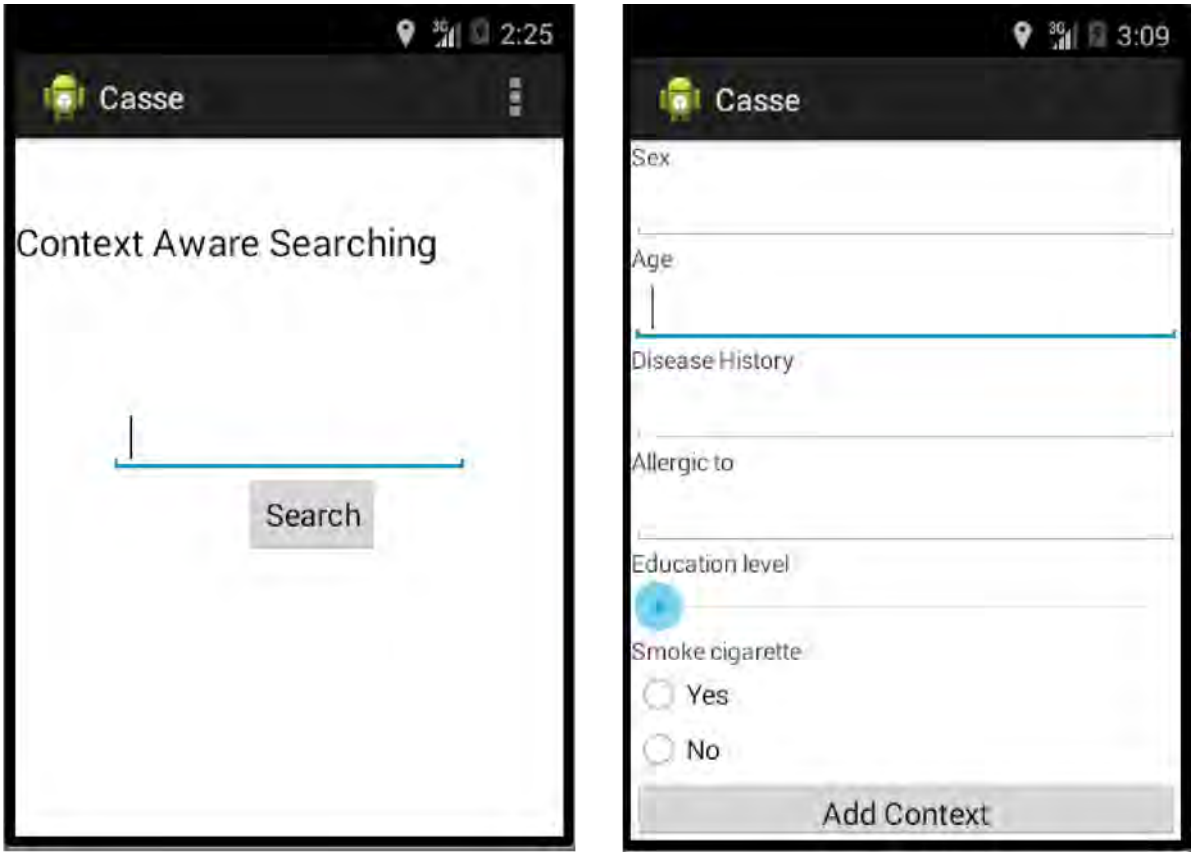


Figure 5-1: Ontology class representation graph for the context representation

5.5 Demonstration

Figure 5-2 shows the client side application interface of the proposed context aware semantic search on the client mobile phone. As it is shown on the *Figure 5-2 a*, the UI consists of text box search button. The textbox is used to type search query. The search button is used to submit the search query. The client application is also used to set preference (static contexts). *Figure 5-2 b* shows preference setting page for health domain.



a) Context aware search interface b) User context acquisition interface

Figure 5-2: Client side application interface

Figure 5-3 shows the result of executing query representing the scenario presented in *Section*5.1. The full result is attached at Annex (c.f. *Sample Result*).

Figure 5-3 :Query result from CASSE

5.6 Evaluation

In order to evaluate our system, an experiment is designed and conducted to test the precision instead of the recall since it does not require the knowledge of the test collections. Precision is the number of relevant documents retrieved divided by the total number of retrieved documents [56]. However, relevance is subjective and different users may have different relevance measures. In our work, the relevancy of the retrieved document to the query is evaluated by six information science post graduate students who are regular users of the Web.

Precision takes all retrieved documents into account, but it can also be evaluated at a given cut-off rank, considering only the topmost results returned by the system. This measure is called precision at n or P@n [56]. For our system we calculate precision by considering top 15 results. Precision for the top 15 results is obtained by taking the ratio of relevant documents retrieved in the first 15 search results by 15. The evaluation result using is presented in *Table 5-4*.

Table 5-4: Precision evaluation result

Keyword	Relevant Retrieved	Total Retrieved	P@15
Cancer	14	15	0.933
Hospital	10	15	0.66
Treatment for cancer	13	15	0.866
psychiatric hospital	7	15	0.466
radiotherapy	10	15	0.66
Average			0.717

As it is shown on *Table 5-4* the precision of our context aware semantic search engine is 71.7 %. Due to difference in datasets [15] we could not make a comparison of the above result with other semantic search engines rather we conduct experiment two to see the advantage of having context parameters. In this case we calculate the precision for the results without context parameters with the same dataset. *Table 5-5* shows the precision comparison of the two results.

Table 5-5 Experiment two precision result

Keyword	RR on Exp1	RR on Exp2	P@15 on Exp1	P@15 on Exp2
Cancer	14	7	0.933	0.466
Hospital	10	8	0.66	0.533
Treatment for cancer	13	10	0.866	0.66
psychiatric hospital	7	6	0.466	0.4
radiotherapy	10	7	0.66	0.466
Average			0.717	0.4118

Where **RR** is relevant document retrieved

[15] Our datasets is limited to dbpedia datasets which is a collection of Wikipedia articles. Other semantic search engines have their own collection of datasets.

Exp1 is using our context aware semantic search engine on DBPedia datasets executing query representing the scenario presented in *Section*5.1.

Exp2 is using only user query on DBPedia datasets

P@15 on Exp1 is precision at top 15 results on Exp1

P@15 on Exp2 is precision at top 15 results on Exp2

Table 5-6 shows the comparison for top 15 query result from experiment one and experiment two.

Table 5-6: comparison of results

Criteria	Result from Exp1	Results from Exp2
Number of result	15	15
Hospital type	All Community	7 community, 3 teaching, 4 private and 1 both teaching and community.
Language	English	English and 2 French and 1 Arabic
Location	All United State	3 United Kingdom 4 United State 1 from Philippines, China 2 France
Order	By location	Unordered
Query time	34 second	15 second

The experimental results in *Table 5-5* and *Table 5-6Table* ***5-5*** showed that our context aware semantic search engine has high precision than keyword base search engines. This clearly shows the advantage of having user context while searching the web.

CHAPTER 6
CONCLUSION AND FUTURE WORK

In this work, we present a study on context aware semantic search. This section discusses the achievements of the research, contribution and possible future works to improve the work carried out by this research.

6.1 Conclusion

Context aware computing enhances service deliveries in various application areas. During the process of the study, we come up with the following conclusions

- If the knowledge in the dataset is incomplete SPARQL queries will return less results than expected, and the relevant documents will not be retrieved,
- User context and device context play vital rule for retrieving more relevant documents.
- The more the application knows about the user's context, the more likely it is that it can deliver documents the user wants.
- Since all contexts are not equally important for every query, context selection should be addressed.

The contribution of this thesis work is summarized as follows:

- Provide architecture for context aware semantic search with capabilities of context acquisition, management, reasoning and querying the semantic web along with a search history.
- Provide a way of integrating users context while searching semantic web
- Provide an algorithm to find the search concept from search history of the user.
- Mapping the search concept with context.
- Provide a way of applying application advantage of pervasive computing into the different application areas.

6.2 Future Works

Finally as a future work, we plan to continue working on the following issues.

- Implementation of this study in non-android based operating system
- Implementation of the study in multiple domains.
- Developing Desktop/web version of the client app to gather additional search history made by the same user other than the his/her smart phone.

REFERENCES

[1] Junaidah Mohamed Kassim and Mahathir Rahmany, "*Introduction to semantic search engine"*.2009 International Conference on Electrical Engineering and Informatics.5-7 August 2009, Selangor, Malaysia

[2] Debajyoti M, Aritra B, Sreemoyee M, JhilikBhattacharya",*Adomain specific ontology based semantic web search engine"*. West Bengal University of technology ,Calcutta 700091.

[3] Berners-Lee, Tim, James Hendler, and Ora Lassila. "The semantic web." *Scientific american* 284.5 (2001): 28-37.Schilit, Adams, & Want "*Context aware computing Applications*."1994.

[4] Schilit, Bill, Norman Adams, and Roy Want. "Context-aware computing applications." In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pp. 85-90. IEEE, 1994.

[5] Microsoft: 53 Percent Of Mobile Searches Have Local Intent http://searchengineland.c om/microsoft-53-percent-of-mobile-searches-have-local-intent-55556. *Retrieved* 2 *September 2014.*

[6] Dey, Anind K. "Understanding and using context." *Personal and ubiquitous computing* 5, no. 1 (2001): 4-7.

[7] Costa, Patricia Dockhorn, Giancarlo Guizzardi, Joao Paulo A. Almeida, Luis Ferreira Pires, and Marten van Sinderen. "Situations in Conceptual Modeling of Context." In *EDOC Workshops, p. 6. 2006.*

[8] Korkea-Aho, Mari. "Context-aware applications survey." *Department of Computer Science, Helsinki University of Technology* (2000).

[9] Gellersen, Hans W., Albercht Schmidt, and Michael Beigl. "Multi-sensor context-awareness in mobile devices and smart artifacts." *Mobile Networks and Applications* 7, no. 5 (2002): 341-351

[10] Schwinger, W., Chr Grün, B. Pröll, W. Retschitzegger, and A. Schauerhuber. "Context-awareness in mobile tourism guides–A comprehensive survey." *Rapport Technique. Johannes Kepler University Linz* (2005).

[11] Samsung Galaxy S5 specs http://www.phonearena.com/phones/Samsung-Galaxy-S5_id8202. *Retrieved* 12 September 2014.

[12] World Wide Web Consortium (W3C) specifications. "HTML 4.01 specification." (1999).

[13] Allen, J., Jay Aslam, Nicholas Belkin, Chris Buckley, Jamie Callan, W. B. Croft, Sue Dumais et al. "Challenges in Information Retrieval and Language Modeling." In *SIGIR Forum*, vol. 37, no. 1, pp. 31-47. ACM Press, 2003.

[14] Jansen, Bernard J., Amanda Spink, Judy Bateman, and Tefko Saracevic. "Real life information retrieval: A study of user queries on the web." In *ACM SIGIR Forum*, vol. 32, no. 1, pp. 5-17. ACM, 1998.

[15] Lee, Uichin, Zhenyu Liu, and Junghoo Cho. "Automatic identification of user goals in web search." In *Proceedings of the 14th international conference on World Wide Web*, pp. 391-400. ACM, 2005.

[16] Technologies behind Google ranking http://googleblog.blogspot.com/2008/07 /technologies-behind-google-ranking.html. *Retrieved 23 April 2014.*

[17] Peshave, Monica, and Kamyar Dezhgosha. "How Search Engines Work: and a Web Crawler Application." PhD diss., University of Illinois Springfield, 2005.

[18] By Yuanlei Zhang, A Comparison of Search Engines For Finding Resources By Yuanlei Zhang, April 28, 2004

[19] Raja Iswary, Keshab Nath. "Web Crawler". *In International Journal of Advanced Research in Computer and Communication Engineering* c Vol. 2, Issue 10, October 2013. Keshab Nath.

[20] Madhu, G., Dr A. Govardhan, and Dr TV Rajinikanth. "Intelligent semantic web search engines: a brief survey." *arXiv preprint arXiv:1102.0831* (2011).

[21] RDF Semantic web framework, http://www.w3.org/RDF/. *Retrieved 9 April 2014.*

[22] W3C Consortium Recommendation*, http://www.w3.org/TR/. Retrieved 19 October 2014.*

[23] Madhu, G., Dr A. Govardhan, and Dr TV Rajinikanth. "Intelligent semantic web search engines: a brief survey." *arXiv preprint arXiv:1102.0831* (2011).

[24] What is ontology and why we need it, http://www.ksl.stanford.edu/people/dlm/papers /ontology101/ontology101-noy-mcguinness.html. *Retrieved 19 April 2014.*

[25] Modularization of Ontologies, http://www.obitko.com/tutorials/ontologies-semantic-web/modularization-of-ontologies.html. *Retrieved 19 April 2014.*

[26] McGuinness, Deborah L., and Frank Van Harmelen. "OWL web ontology language overview." *W3C recommendation* 10, no. 10 (2004): 2004.

[27] W3C SPARQL Working Group. "SPARQL 1.1 overview." *World Wide Web Consortium, Recommendation REC-sparql11-overview-20130321* (2013).

[28] Hakia, http://company.hakia.com/whatis.html. *Retrieved 23 April 2014.*

[29] Mäkelä, Eetu. "Survey of semantic search research." In *Proceedings of the seminar on knowledge management on the semantic web*. Department of Computer Science, University of Helsinki, Helsinki, 2005.

[30] Jones, Gareth JF, and Peter J. Brown. "Context-aware retrieval for ubiquitous computing environments." In *Mobile and ubiquitous information access*, pp. 227-243. Springer Berlin Heidelberg, 2004.

[31] Chen, Jidong, Wentao Wu, Hang Guo, and Wei Wang. "Context-aware Search for Personal Information Management Systems." In *SDM*, pp. 708-719. 2012.

[32] Jiang, Di, Kenneth Wai-Ting Leung, and Wilfred Ng. "Context-aware search personalization with concept preference." In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 563-572. ACM, 2011.

[33] Cao, Huanhuan, Daxin Jiang, Jian Pei, Enhong Chen, and Hang Li. "Towards context-aware search by learning a very large variable length hidden markov model from search logs." In *Proceedings of the 18th international conference on World wide web*, pp. 191-200. ACM, 2009.

[34] Sriram, Smitha, Xuehua Shen, and Chengxiang Zhai. "A session-based search engine." In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 492-493. ACM, 2004.

[35] Wang, Haodong, Chiu C. Tan, and Qun Li. "Snoogle: A search engine for pervasive environments." *Parallel and Distributed Systems, IEEE Transactions on* 21, no. 8 (2010): 1188-1202.

[36] Loyall, Joseph P., and Richard E. Schantz. "Using context awareness to improve quality of information retrieval in pervasive computing." In *Software Technologies for Embedded and Ubiquitous Systems*, pp. 320-331. Springer Berlin Heidelberg, 2009.

[37] Xiang, Biao, Daxin Jiang, Jian Pei, Xiaohui Sun, Enhong Chen, and Hang Li. "Context-aware ranking in web search." In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pp. 451-458. ACM, 2010.

[38] Hong, Jong-yi, Eui-ho Suh, and Sung-Jin Kim. "Context-aware systems: A literature review and classification." *Expert Systems with Applications* 36, no. 4 (2009): 8509-8522.

[39] Yonas Tadess, “Architecture for Context Management and Context-Aware Service Delivery ” Master’s thesis, school of information science, Addis Ababa University. May, 2014

[40] Anteneh Alemu, “Context Aware Mobile Marketing: The case of markets in Ethiopia” Master’s thesis, school of information science, Addis Ababa University. June 2014

[41] Bethelhem Teka, “Modeling Pervasive Context-aware Museum Guide Service (PCMGS)”. Master’s thesis, school of information science, Addis Ababa University. June 2014

[42] Graupmann, Jens, and Ralf Schenkel. "GeoSphereSearch: Context-Aware Geographic Web Search." In *GIR*. 2006.

[43] Yokoji, Seiji, Katsumi Takahashi, and Nobuyuki Miura. "Kokono Search: A Location Based Search Engine." In *WWW Posters*. 2001.

[44] Watters, Carolyn, and Ghada Amoudi. "GeoSearcher: Location-based ranking of search engine results." *Journal of the American Society for Information Science and Technology* 54, no. 2 (2003): 140-151.

[45] Guha, Ramanathan, Rob McCool, and Eric Miller. "Semantic search." In *Proceedings of the 12th international conference on World Wide Web*, pp. 700-709. ACM, 2003.

[46] Buntine, Wray L., and Michael P. Taylor. "ALVIS: Superpeer Semantic Search Engine." In *EWIMT*. 2004.

[47] Celino, Irene, E. D. Valle, D. Cerzza, and Andrea Turati. "Squiggle: a semantic search engine for indexing and retrieval of multimedia content." *Proceedings of SAMT* 2006 (2006): 20-34.

[48] Al-Rawi, Salah Sleibi, Ahmed Tariq Sadiq, and Sumaya Abdulla Hamad. "Design and Evaluation of Semantic Guided Search Engine." *International Journal of Web Engineering* 1, no. 3 (2012): 15-23.

[49] Hakia - First Meaning-based Search Engine, http://readwrite.com/2006/12/07/hakia_meaning-based_search Retrieved 1 December 2014

[50] Rajasurya, Swathi, Tamizhamudhu Muralidharan, Sandhiya Devi, and S. Swamynathan. "Semantic information retrieval using ontology in university domain." *arXiv preprint arXiv:1207.5745* (2012).

[51] Aung, Nwe Ni, and Thinn Thu Naing. "Sports Information Retrieval with Semantic Relationships of Ontology." *International Proceedings of Economics Development & Research* 12 (2011).

[52] Lei, Yuangui, Victoria Uren, and Enrico Motta. "Semsearch: A search engine for the semantic web." In *Managing Knowledge in a World of Networks*, pp. 238-245. Springer Berlin Heidelberg, 2006.

[53] Sensor, http://en.wikipedia.org/wiki/Sensor Retrieved 12 May 2014.

[54] Church, Karen, and Barry Smyth. "Understanding the intent behind mobile information needs." *Proceedings of the 14th international conference on Intelligent user interfaces*. ACM, 2009.

[55] Semantic Matching, http://semanticmatching.org/s-match.html. Retrieved 23 September 2014.

[56] Precision and Recall, http://en.wikipedia.org/wiki/Precision_and_recall, Retrieved on December 23 2015.

[57] Aréchiga, Daniel, Jesús Vegas, and Pablo De la Fuente Redondo. "Ontology supported personalized search for mobile devices." *Proceedings of the ONTOSE* (2009).

ANNEXES

A. Sample Result

|http:/en.wikipedia.org/wiki/Maimonides_Medical_Center> | "Maimonides Medical Center is a non-profit, non-sectarian hospital located in Borough Park, in the New York City borough of Brooklyn, in the U.S. state of New York. Maimonides is both a treatment facility and academic medical center with 711 beds, and more than 70 primary care and sub-specialty programs. With a staff of nationally renowned physicians, Maimonides Medical Center strives to conduct quality research, care and education."@en
|

| <http://en.wikipedia.org/wiki/MedStar_Washington_Hospital_Center> | "MedStar Washington Hospital Center is the largest private hospital in Washington, D.C. A member of MedStar Health, the not-for-profit Hospital Center is licensed for 926 beds. Health services in primary, secondary and tertiary care are offered to adult and neonatal patients. It also serves as a teaching hospital for Georgetown University School of Medicine.The Hospital Center occupies a 47-acre (19 ha) campus in Northwest Washington that it shares with three other medical facilities. Immediately adjacent to MedStar Washington Hospital Center are the National Rehabilitation Hospital and the central branch of Children's National Medical Center."@en
|

| <http://en.wikipedia.org/wiki/John_H._Stroger,_Jr._Hospital_of_Cook_County> | "The John H. Stroger, Jr. Hospital of Cook County, formerly Cook County Hospital is a public urban teaching hospital in Chicago that provides primary, specialty and tertiary healthcare services to the five million residents of Cook County, Illinois. The hospital has a staff of 300 attending physicians along with more than 400 medical residents and fellows. The hospital campus, located at 1901 W. Harrison Street Chicago, Illinois, is a part of the 305-acre (123 ha) Illinois Medical District, which is one of the largest concentrations of medical facilities in the world. The hospitalâ€™s 1,200,000 square feet (110,000 m2) represent the equivalent of 25 football fields. The layout of the facility organizes services in a â€œmain street-styleâ€? to accommodate the needs of patients, physicians and staff. The hospital offers dedicated units for obstetrics and pediatrics, intensive care, and burns. It boasts one of the most respected emergency rooms in the country and a Level 1 Trauma Center. The Adult ER treats over

110,000 patients annually, while the Pediatrics ER treats 45,000 children and adolescents each year.@en

|

| <http://en.wikipedia.org/wiki/CentraState_Medical_Center> | "CentraState Healthcare System, also known as CentraState Medical Center, is a non-for-profit community health organization consisting of an acute-care hospital, an ambulatory care campus, three senior living communities, a Family Medicine Residency Program, and a charitable foundation. It is a member of the Rutgers Robert Wood Johnson Health Network.Established in 1971, CentraState Healthcare System consists of the Main Medical Center, three senior living communities, an ambulatory campus, a wellness & fitness center, and a family medicine center. The Main Medical Center building has 282 beds.CentraStateâ€™s mission is to enhance the health and well-being of the community through the compassionate delivery of quality healthcare. CentraState is affiliated with the Robert Wood Johnson Medical School, Robert Wood Johnson Health Network."@en

|

| <http://en.wikipedia.org/wiki/James_Cancer_Hospital> | "The Ohio State University Comprehensive Cancer Center - Arthur G. James Cancer Hospital and Richard J. Solove Research Institute (commonly shortened to just James Cancer Hospital, officially abbreviated OSUCCC - James, and colloquially referred to as The James) is part of The Ohio State University and one of the twenty-one National Comprehensive Cancer hospitals. It is named after Arthur G. James, the founder, who desired a cancer hospital in Columbus, Ohio, United States. The hospital conducts treatments for cancer, and conducts research in the Solove Research Institute. The James receives donations through the Pelotonia biking event. Ohio Stateâ€™s James Cancer Hospital recently received the highest nursing recognition with the prestigious MagnetÂ® designation."@en

|

| <http://en.wikipedia.org/wiki/Regions_Hospital> | "Regions Hospital is a teaching hospital located in St. Paul, Minnesota, part of the HealthPartners system. The hospital is an ACS verified Level I Trauma Center for both children and adults, and was Minnesota's first pediatric level one trauma center. Regions Hospital is a leading, full-service, private hospital, with special programs in heart, cancer, behavioral health, burn, emergency and trauma."@en

B. Sample Code

```
//
import android.util.Log;
import android.view.LayoutInflater;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.hp.hpl.jena.query.ARQ;
import com.hp.hpl.jena.query.DatasetFactory;
import com.hp.hpl.jena.query.Query;
import com.hp.hpl.jena.query.QueryExecution;
import com.hp.hpl.jena.query.QueryExecutionFactory;
import com.hp.hpl.jena.query.QueryFactory;
import com.hp.hpl.jena.query.QuerySolution;
//*;
//import com.hp.hpl.jena.vocabulary.ResultSet;
import com.hp.hpl.jena.query.ResultSet;
import com.hp.hpl.jena.query.ResultSetFormatter;
import com.hp.hpl.jena.query.Syntax;
import com.hp.hpl.jena.sparql.engine.http.QueryEngineHTTP;
public class QueryDBpedia {
    public static void main(String[] args) {
        QueryDBpedia queryDBpedia = new QueryDBpedia();
        queryDBpedia.queryExternalSources();
   }
public void queryExternalSources() {
```

```
 String sparqlQueryString = " PREFIX owl:
<http://www.w3.org/2002/07/owl#> "
+ "PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> "
+ "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> "
+ "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> "
  + "PREFIX foaf: <http://xmlns.com/foaf/0.1/> "
  + "PREFIX dc: <http://purl.org/dc/elements/1.1/>"
  + "PREFIX : <http://dbpedia.org/resource/> "
  + "PREFIX dbpedia2: <http://dbpedia.org/property/> "
  + "PREFIX dbpedia: <http://dbpedia.org/>"
  + "PREFIX skos: <http://www.w3.org/2004/02/skos/core#>"
  + "PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> "
  + "PREFIX dbo: <http://dbpedia.org/ontology/>  "
  + "PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>  "
  + "PREFIX dcterms: <http://purl.org/dc/terms/> "
  + "PREFIX spatial: <http://jena.apache.org/spatial#> "
  + "SELECT DISTINCT ?wikilink ?abstract"
  + " WHERE {"
          + " ?Hospital a dbpedia-owl:Hospital."
          + " ?Hospital dbpedia2:name ?name."
          + " ?Hospital dbpedia2:beds ?bedcount."
          + "?Hospital geo:lat ?latitude."
          + "?Hospital geo:long ?longitude."
          +" ?Hospital dbpedia2:country ?country."
          +"?Hospital dbpedia2:type ?type."
          +"?Hospital dbpedia-owl:abstract ?abstract."
          + "?Hospital foaf:isPrimaryTopicOf ?wikilink . "
          +"FILTER ( (?longitude > (longi - 100)) &&
(?longitude < (longi + 100)) && (?latitude > (lati - 100)) &&
(?latitude < (lati + 100)) && regex(str(?type), type,
\"i\"))\n"
// spatial:nearby (?latitude, ?longitude, 100)"
 +  "  }\n" +
                "LIMIT  15 \n" +
```

```
                    "";
          Query query = QueryFactory.create(sparqlQueryString);
         ARQ.getContext().setTrue(ARQ.useSAX);
        //Executing SPARQL Query and pointing to the DBpedia
SPARQL Endpoint
         QueryExecution qexec =
QueryExecutionFactory.sparqlService("http://dbpedia.org/sparql
", query);
        //Retrieving the SPARQL Query results
         ResultSet results = qexec.execSelect();
         ResultSetFormatter.out(System.out, results, query) ;

         qexec.close();

    }
```

```
//Static context class
package com.example.casse;
import android.app.Activity;
import android.database.Cursor;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class Scontexts extends Activity  {
        DbAdapter db= new DbAdapter(this);
        @Override
        protected void onCreate(Bundle savedInstanceState) {
                // TODO Auto-generated method stub
                super.onCreate(savedInstanceState);
                setContentView(R.layout.scontext);
                db.open();
```

```
                Cursor c=db.getAllRecords();
                if(c.moveToFirst())
                {
                        do{
                                DisplayRecord(c);
                         } while (c.moveToNext());
                }
               //db.close();
        }
        public void Addcontext(View v){
                Log.d("context ","Adding ");
                EditText prof =(EditText)findViewById(R.id.editProf);
                EditText age =(EditText)findViewById(R.id.editAge);
    db.open();
                long id=db.insertRecord(prof.getText().toString(),age.getText().toString());
                prof.setText("");
                age.setText("");
                Toast.makeText(this,"Context Added", Toast.LENGTH_LONG).show();
        }
        public void DisplayRecord(Cursor c)
        {
                        Toast.makeText(this,
                                        "id: " + c.getString(0) + "\n" +
                                        "Profession: " + c.getString(1) + "\n" +
                                        "Age : " + c.getString(2),
                                        Toast.LENGTH_SHORT).show();

        }

}
```

C. Sample Ontology

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
<!ENTITY owl "http://www.w3.org/2002/07/owl#" >
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>
<rdf:RDF
xmlns=http://www.semanticweb.org/ted/ontologies/2014/7/untitled-
ontology-4#
ml:base="http://www.semanticweb.org/ted/ontologies/2014/7/untitle
d-ontology-4"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<owl:Ontology
rdf:about="http://www.semanticweb.org/ted/ontologies/2014/7/untit
led-ontology-4"/>
<!--
     /////////////////
     //
     // Object Properties
     ///////////////////
<!-- http://www.semanticweb.org/ted/ontologies/2014/7/untitled-
ontology-4#Has_Gender -->
<owl:ObjectProperty
rdf:about="http://www.semanticweb.org/ted/ontologies/2014/7/untit
led-ontology-4#Has_Gender"/>


<!-- http://www.semanticweb.org/ted/ontologies/2014/7/untitled-
ontology-4#Has_latitude --><owl:ObjectProperty
rdf:about="http://www.semanticweb.org/ted/ontologies/2014/7/untit
led-ontology-4#Has_latitude">
```

```
    <rdfs:domain
    rdf:resource="http://www.semanticweb.org/ted/ontologies/2014/7/un
    titled-ontology-4#Location"/>
    </owl:ObjectProperty>
<!-- http://www.semanticweb.org/ted/ontologies/2014/7/untitled-
ontology-4#Has_longitude -->
    <owl:ObjectProperty
    rdf:about="http://www.semanticweb.org/ted/ontologies/2014/7/untit
    led-ontology-4#Has_longitude">
    <rdfs:domain
    rdf:resource="http://www.semanticweb.org/ted/ontologies/2014/7/un
    titled-ontology-4#Location"/>
    </owl:ObjectProperty>
    <!-- http://www.semanticweb.org/ted/ontologies/2014/7/untitled-
    ontology-4#has_address -->

    <owl:ObjectProperty
    rdf:about="http://www.semanticweb.org/ted/ontologies/2014/7/untit
    led-ontology-4#has_address"/>
    <!-- http://www.semanticweb.org/ted/ontologies/2014/7/untitled-
    ontology-4#has_location -->

    <owl:ObjectProperty
    rdf:about="http://www.semanticweb.org/ted/ontologies/2014/7/untit
    led-ontology-4#has_location">
    <rdfs:domain
    rdf:resource="http://www.semanticweb.org/ted/ontologies/2014/7/un
    titled-ontology-4#Place"/>
    </owl:ObjectProperty>



    <!-- http://www.semanticweb.org/ted/ontologies/2014/7/untitled-
    ontology-4#has_name -->
```

```
<owl:ObjectProperty
rdf:about="http://www.semanticweb.org/ted/ontologies/2014/7/untit
led-ontology-4#has_name">
<rdfs:domain
rdf:resource="http://www.semanticweb.org/ted/ontologies/2014/7/un
titled-ontology-4#Activity"/>
<rdfs:domain
rdf:resource="http://www.semanticweb.org/ted/ontologies/2014/7/un
titled-ontology-4#Person"/>
</owl:ObjectProperty>

<!-- http://www.semanticweb.org/ted/ontologies/2014/7/untitled-
ontology-4#hasemail -->
<owl:ObjectProperty
rdf:about="http://www.semanticweb.org/ted/ontologies/2014/7/untit
led-ontology-4#hasemail">
<rdfs:domain
rdf:resource="http://www.semanticweb.org/ted/ontologies/2014/7/un
titled-ontology-4#Person"/>
</owl:ObjectProperty>
<!--

///////////////////////////////////////////////////////////////////
//////////////////////
    //
    // Data properties
    //

///////////////////////////////////////////////////////////////////
//////////////////////
     -->
<!-- http://www.semanticweb.org/ted/ontologies/2014/7/untitled-
ontology-4#Fullname -->

<owl:DatatypeProperty
rdf:about="http://www.semanticweb.org/ted/ontologies/2014/7/untit
led-ontology-4#Fullname">
```

```
<rdfs:domain>
<owl:Restriction>
<owl:onProperty
rdf:resource="http://www.semanticweb.org/ted/ontologies/2014/7/un
titled-ontology-4#Fullname"/>
<owl:someValuesFrom rdf:resource="&xsd;string"/>
</owl:Restriction>
</rdfs:domain>
</owl:DatatypeProperty>



<!-- http://www.semanticweb.org/ted/ontologies/2014/7/untitled-
ontology-4#Gender -->

<owl:DatatypeProperty
rdf:about="http://www.semanticweb.org/ted/ontologies/2014/7/untit
led-ontology-4#Gender">
<rdfs:domain>
<owl:Restriction>
<owl:onProperty
rdf:resource="http://www.semanticweb.org/ted/ontologies/2014/7/un
titled-ontology-4#Gender"/>
<owl:someValuesFrom rdf:resource="&xsd;string"/>
</owl:Restriction>
</rdfs:domain>
</owl:DatatypeProperty>


  // Classes
<!-- http://www.semanticweb.org/ted/ontologies/2014/7/untitled-
ontology-4#Activity -->

<owl:Class
rdf:about="http://www.semanticweb.org/ted/ontologies/2014/7/untit
led-ontology-4#Activity"/>
```

```
<!-- http://www.semanticweb.org/ted/ontologies/2014/7/untitled-
ontology-4#Building --><owl:Class
rdf:about="http://www.semanticweb.org/ted/ontologies/2014/7/untit
led-ontology-4#Building">
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ted/ontologies/2014/7/un
titled-ontology-4#Place"/>
</owl:Class>
<!-- http://www.semanticweb.org/ted/ontologies/2014/7/untitled-
ontology-4#Campus -->

<owl:Class
rdf:about="http://www.semanticweb.org/ted/ontologies/2014/7/untit
led-ontology-4#Campus">
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ted/ontologies/2014/7/un
titled-ontology-4#Place"/>
</owl:Class>
```

D. Declaration

I, the undersigned, declare that this research is my original work and has not been presented for degree in any other university, and that all sources of materials used for the research have been acknowledged.

Declared by:

Name: Tewodros Worku

Signature: _____

Date: _____

Confirmed by advisor:

Name: Fekade Getahun (PhD)

Signature: _____

Date: _____