ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
COLLEGE OF NATURAL SCIENCES
DEPARTMENT OF COMPUTER SCIENCE

Mobile Based Crowdsourcing for Agricultural Market Information

Natnael Alemayehu

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OF THE ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE

FEBRUARY, 2015

Addis Ababa

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
COLLEGE OF NATURAL SCIENCE
DEPARTMENT OF COMPUTER SCIENCE

Mobile Based Crowdsourcing for Agricultural Market Information

Natnael Alemayehu

ADVISOR:
Mulugeta Libsie (PhD)

APROVED BY

EXAMINING BOARD:

1. **Dr. Mulugeta Libsie, Advisor** _____
2. **Dr. Dida Midekso, Examiner** _____
3. **Dr. Solomon Atnafu, Examiner** _____

Dedicated to

My Dear Mother

Acknowledgments

First of all, I would like to thank God, for all his blessings and helping me pass all the challenges and those hard times that I couldn‘t pass by my own.

I owe my deepest gratitude to my advisor Dr Mulugeta Libsie for accepting this thesis work and dedicating his time, patience and also providing comments all the way through this study. His eagerness and encouragement has always inspired me towards the completion of the work.

I would also like to thank the Department of Computer Science and all my instructors, for their personal commitment and contribution to the success of the graduate program.

Lastly, I would also like to express my appreciation and thanks to my family, friends and colleagues who have helped me in so many ways.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Abstract

The availability of market information is vital for individuals and countries' economic growth. By its nature market information is dynamic and its source is distributed among the public and must be communicated so that all stakeholders get access to it. In Ethiopia's case, agricultural information, especially agro-market information, is important due to the fact that Ethiopia depends on agriculture. Sadly, there is an information gap between the various stakeholders, namely the farmer and the business man regarding market information.

In order to overcome such problem, we have proposed and implemented a mobile based crowdsourcing system for agro market information. The aim of our proposed system is to process and provide aggregated and updated agricultural market price information which is provided by the crowd using SMS messaging that is available in Java based mobile phone devices. Our proposed system has five basic components which are: SMS Manager Component, Information Extraction component, Data analysis component, Crowd Management and Request Management component.

As a means of evaluating the proposed architecture, we developed a prototype that implements the components of the proposed architecture. We also evaluated the implementation by testing sample SMS messages. Results from the evaluation showed that the proposed architecture can provide a mobile based crowdsourcing system for agro market information for low end mobile phone users with price aggregation performance of 72%.

Key words: Crowdsourcing, Agricultural Market Information, Ontology, Information Extraction, Mobile Based Crowdsourcing

Chapter One: Introduction

1.1 Introduction

Humans communicate to share information and get new knowledge. They use natural language to communicate with one another. One of the areas where they communicate is trade. In trade ventures, people exchange and share market information. If they need information that is at far distance they rely on various technologies like mobile phones.

The availability of market information is vital for individuals and countries' economic growth. By its nature market information is dynamic and its source is distributed among the public and must be communicated so that all stakeholders get access to it. Crowdsourcing can play a vital role in collecting market information from the general public and provide it back to the society.

Howe [2] defined crowdsourcing as "*an act of taking a job traditionally performed by a designated agent and outsourcing it to an undefined, generally large group of people in the form of an open call".*

Crowdsourcing is an online, distributed problem-solving and production model that has emerged in recent years [3]. It is applicable in many domains such as natural language processing [4], business [5], learning [6], and scientific problem-solving [7]. Some of the well-known models and services that use crowdsourcing are:

- ✓ *Wikipedia* [8]: a platform that provides a collaborative document editing and knowledge sharing.
- ✓ *Ushahidi*[9]: a mapping tool for crowdsourcing crisis information. It enables the public to communicate crisis information via SMS, e-mail, or web entry. It has been used to manage relief efforts in Haiti during the earthquake crisis in 2010.
- ✓ *Threadless*[10]: a web-based t-shirt company that crowdsources the design process for its shirts through an ongoing online competition.
- ✓ *txteagle* [11]: a mobile based system enabling people to earn small amount of money by completing simple tasks (micro tasks) on their mobile phone for corporations. The tasks involve providing information for surveys and local language translation.
- ✓ mCollect [12]: a market information collection system, is part of a broader initiative which also includes Market Prices and Market Alerts. It has been implemented in Burkina Faso, Mali, Senegal and Liberia.

One of the successful implementation domains of crowdsourcing is collecting large volume of information or knowledge from a crowd and providing them to the public using ICT [13]. One of the problems in Ethiopia regarding information sharing is the lack of ICT support. This problem is magnified in agricultural marketing where the ICT support is almost non-existent. Therefore, we propose that designing and developing a crowdsourcing service can be a solution for this problem.

1.2 Motivation

Crowdsourcing is gaining more and more popularity in the research community, regardless of the fact that it is a recent concept in information technology [13]. This is because it is a way of collaboratively doing tasks or solving problems. In the developed countries, people are using crowdsourcing to be successful in their business areas (like market analyses and branding) and also solving their problems such as natural language processing (e.g., collaborative translation [4]) and sharing of knowledge (e.g., Wikipedia [8]).

It is time for developing countries like Ethiopia to share the success and power of crowdsourcing in solving problems in different areas, especially in the agro-market area which is becoming very hard and competitive for the farmers in the rural areas that don't have much access to ICT. The following scenario will illustrate this concept

Scenario

When farmers produce an agricultural product and want to sell it, they have to take their products to the local market and directly sell it to the customers or market brokers who will buy their product and take it to other markets. As many farmers live in rural areas, they only have access to the local market price and they have to rely on what others told them about market prices in other market places, especially about prices in urban markets. Due to this, unknowingly they will be forced to sell their products in cheap and unfair prices to the brokers that will sell the product in other markets with a large profit. If there had been a way or a means for the farmers to have the aggregated market price information for their product in the general market, they could have sold their products in a fair price.

What we can see from the above scenario is that, due to lack of market information in rural areas, farmers are not getting what they deserve for their product and have continued to rely on market information supplied and verified through traditional word-of-mouth approach.

We can also see the need for a service or a technological system in which it is possible to communicate market information in a way suitable and affordable to farmers in rural areas where there is no computer or Internet connection but only low-end mobile devices.

Amanuel Zewge *et al*. [1] have come up with a framework for supporting the agricultural sector with an ICT support in all its areas in Ethiopia. They have also suggested that there is a need for ICT support in the agro-marketing sector which is open for research. Therefore conducting a crowdsourcing research to provide a support in this area is a big contribution for Ethiopia and could help farmers in rural areas to have access and to contribute for collaborative real time market information.

1.3 Statement of the Problem

Agriculture-led development is the priority sector of most African countries [1]. In Ethiopia's case, agricultural information, especially agro-market information, is important due to the fact that Ethiopia depends on agriculture. Sadly there is an information gap between the various stakeholders, namely the farmer and the business man regarding market information. Even if there are central bodies that are responsible for collecting and providing market information in various levels, they lack coordination and ICT support to share the market information among themselves and with the other stakeholders [1]. The organization that is working hard to curb this problem is the ECX.

The Ethiopian Commodity Exchange(ECX), an authority that provides market prices for commodities like Coffee and oil-seed products, plays a great role in bringing together the stakeholders and minimizing the aforementioned information gap[1].

The ECX, developed by Dr Elani G/medhin, provides an ICT supported system to the users and is able to provide a fair market price for the farmer and the businessman by collecting market prices and collecting the products from the farmers in their satellite offices and biding the products on the exchange system and thus, allowing the farmers to sell the product by the bided value. Products along with their bid prices are transmitted live on sonic screens that are found in various regions of Ethiopia.

Even though ECX is playing a vital role in providing market information to the general public, it has limitations due to the fact that it only concentrates in few products and the price information's are only available in places that the sonic screen is available.

As market information is dynamic, distributed and its source is the public, in order to communicate it or share it we must have to have a real time collaborative distributed, multi-lingual and cheap information communication technology.

Hence, the problems to be addressed by this research are:

- How to improve the domestic price capture mechanism and increase the level of price co-ordination and trade in Ethiopian rural areas through the available ICT?
- How to model the existing agricultural market information exchange between the various stakeholders?
- What kind of information communication technology can be used to fill the collaboration and sharing gap that exists in the agricultural market information?
- How to design crowdsourcing architecture for low-end mobile systems that can be used to access multi-lingual agricultural market information through collaborative sharing?

1.4 Objectives

General Objective

The general objective of this thesis is to design architecture for a mobile based crowdsourcing platform that provides agro market price gathering, analysis and dissemination.

Specific Objectives

The specific objectives of the research are:

- ✓ Modeling the existing agricultural market information sharing so that we can capture the existing manual agro-market information exchange and use it to model and automate it in the crowdsourcing architecture design.
- ✓ Studying existing crowdsourcing platforms.
- ✓ Developing a bilingual automatic data (text) Extractor and financial data analyzer for the prototype system
- ✓ Developing a prototype for the proposed architecture
- ✓ Evaluating the prototype

1.5 Methodology

In undertaking this research, the following methodologies will be used:

- **Literature Review**

This research will be conducted by first reviewing a number of related literatures and various crowdsourcing services so that it will be easier to choose the appropriate tools and platforms that are required to develop the crowdsourcing architecture.

- **Studying and assessing existing services**

In order to develop the crowdsourcing architecture, we will study the architecture of the existing crowdsourcing services and assess their rating algorithms, data analyzers and interfaces. We will also study the existing agro-market information sharing mechanism and model it. This will help us to come up with and propose a new crowdsourcing architecture that fits to our context and need.

- **Developing a prototype**

Based on the proposed architecture, we will develop a prototype crowdsourcing service. The service will provide its users the ability to share and access agro-market information on their mobile phones using an SMS text that is written in natural language, in our case Amharic or English.

In developing the prototype, Java programming language will be used to develop the crowd management interface, which is responsible for collecting and gathering the incoming and outgoing SMS texts. The database needed for the data will be handled by a dynamic crowd database that is capable of handling crowd database queries. In order to rate and analyze the natural language data gathered from the crowd, a combination of ontological(semantic) approach and statistical analysis will be employed.

- **Testing and Evaluation**

As it is a must to test any system or service, we will conduct a test on the prototype that will be developed based on the architecture designed. In order to test the prototype, a pilot test will be applied to test if the architecture fully models the existing agro-market information sharing and if the problem of the information gap is addressed by the crowdsourcing approach. The

correctness of the linguistic and data analysis algorithms will also be evaluated by the pilot testing.

1.6 Scope and Limitations

Even if there are many sectors that require information sharing regarding agriculture, the scope of the thesis is to model the sharing of agricultural marketing information, which mainly focuses on collection and analysis of market prices for agricultural products, and designing a mobile based architecture for crowdsourcing service.

While developing the prototype for the crowdsourcing service, we will be limited to focusing on two languages, Amharic and English.

1.7Application of Results

Some of the significances of the research are:

- ✓ Bridging the aforementioned agro-market information sharing gap between the various stakeholders.
- ✓ The proposed service will provide better and fair market prices for the stakeholders.
- ✓ Having a service for collaboratively sharing agro-market information using mobile phones.
- ✓ Implementing and localizing crowdsourcing mechanism for solving Ethiopia's problems.
- ✓ It can also be used as a base for other crowdsourcing researches.
- ✓ The service that will be developed using the proposed architecture, will allow sharing of agricultural market information between the various stakeholders (farmers, market brokers, exporters, sellers or anybody interested) through their mobile phone SMS text service.

1.8 Organization of the Thesis

This thesis is organized as follows. Chapter Two covers literature review. Chapter Three presents the existing works that are related to this thesis. Chapter Four discusses the design of the proposed Mobile based Crowdsourcing for agricultural market information system. In Chapter Five we discuss on the implementation of the proposed system. The test and experiment of the system are discussed in Chapter Six. Finally, the conclusion made on the thesis result, the contribution of this research work and recommendation on possible future work related to the thesis are presented in Chapter Seven.

Chapter Two: Literature Review

2.1 Crowdsourcing

Humans perform various tasks and solve problems in order to get some results. To perform these tasks humans have developed various tools. Computer is one of the greatest inventions developed by human as a computational tool. After the invention of computers, humans have tried to automate their tasks and problem solutions so that they can easily be done by computer programs. Sadly, many tasks that are trivial for humans continue to challenge even the most sophisticated computer programs [14]. These problem forces the idea of crowdsourcing to be proposed.

In 2003, Ahn and his colleagues pioneered the concept of "human computation", which employs human abilities to perform computation tasks that are difficult for computers to process [5]. Later, the term "crowdsourcing" was coined by Howe in an article for wired magazine in 2006 [2].

Simply put, crowdsourcing is a distributed problem-solving and business production model in which tasks are broadcasted to a crowd in the form of open calls for solutions. It is distributed in a sense that the solution to the problem or task is going to be aggregated or summarized from the contribution of large and distributed crowd of humans. It is one specific form of harvesting wisdom of the crowd and contributions from users for performing a given task.

The main idea behind crowdsourcing is the wisdom of the crowd. In his book [13], Surowiecki explains wisdom of the crowd as being the collective knowledge, resource and creative power of a group of individuals. According to Surowiecki, under the right circumstances, groups can actually make better decisions than even the smartest person within them. This can be paraphrased as when performing a task or coming up with a solution, a group of people can perform the task or produce a better solution than a single individual or few individuals.

Crowdsourcing has been proved to be effective in areas where the tasks can be easily described to humans and where these tasks are easier to do for humans than for computers [15]. The characteristic of crowdsourcing tasks is that they are typically difficult to solve by computers and easily accomplished by humans. Such tasks like image annotation and data gathering and analysis require the combined effort of human labor and computer's computational power.

When offering tasks to be solved by crowd workers, the person offering the task (*seeker*) defines the product or result that is requested. Typically the seeker has little or no information about the potential crowd workers (*solvers*) that will carry out the task. Hence the description of the task is most critical to achieve the desired result [15].

Many scholars have emphasized the potential for crowdsourcing and related practices to bring about social welfare gains by facilitating innovation, distributed problem solving and access to information resources.

As a strategic model, crowdsourcing tries to attract interested and motivated crowds capable of providing the required solutions in return for incentives (mainly small amounts of money). Often, such so-called crowd workers gather in online communities consisting of experts, small businesses, and other volunteers working in their spare time [15].

2.2 Types of Crowdsourcing

According to Gupta *et al.* [63], recently there are a number of crowdsourcing systems and applications which are developed for accomplishing a certain task or solving a given problem. Crowdsourcing applications and systems can be categorized based on two main factors. The first one is based on the type of platform used to develop the crowdsourcing application and the second category is based on the type of application the crowdsourcing system is to be used.

2.2.1 Crowdsourcing Based on the Platform

Crowdsourcing applications based on the platform on which they would be implemented can be grouped into two; web based and mobile based.

A) Web Based Crowdsourcing

Because of the popularity of Web 2.0 technology, crowdsourcing websites attract much attention at present [16]. Currently there are several websites available that are based on the concept of crowdsourcing and they are more popular compared to their mobile based counterparts [15].

The Internet has become an essential platform for seeking and sharing information, communication, presentation, and collaborating for many users. This is facilitated by many applications, platforms, and services that are provided on the Internet [14]. For many of these

systems, it is essential that Web users actively participate in generating content and providing services. This forces the creation and implementation of various crowdsourcing websites.

A crowdsourcing site has two groups of users: requesters and workers. The crowdsourcing site exhibits a list of available tasks, associating with reward and time period, that are presented by requesters; and during the period, workers compete to provide the best submission. Meanwhile, a worker selects a task from the task list and completes the task because the worker wants to earn the associated reward.

According to Hester *et al.* [18], the spread of networked, digital ICTs have facilitated a rapid expansion in the tools, techniques and applications for crowdsourcing and human computation. Several online labor markets and commercial crowdsourcing service providers like Amazon.com's Mechanical Turk

(a) Contribute to the growth of the tools, techniques, and applications for crowdsourcing, and
(b) Constitute an emergent sector within the ICT industry.

Due to the rapid growth and expansion of Internet, users all around the world, Web based crowdsourcing applications have become a great way to harness the knowledge, skill and information of a large and distributed crowd of people. Nowadays, there are thousands of crowdsourcing applications on the Internet. For example, some of the most known web based crowdsourcing applications are:

- ✓ *Amazon's Mechanical Turk [19]***:** introduced "human intelligence tasks" that can be completed by individuals with a personal computer connected to the Internet for small amounts of money. It is a web service that enables anyone to post 'human intelligence tasks' (HITs) to the AMT's users (commonly known as 'turkers'), as well as be paid for completed tasks.

  Typical HITs include tasks that are easy for humans but extremely difficult for computers such as image tagging, natural language processing, and even survey responses. It is one of the most popular crowdsourcing applications on the Internet and widely used by many turkers as a source of income.

- ✓ *Wikipedia [8]:* online encyclopedias that are written by Internet users and the writing are distributed in that essentially almost anyone can contribute to the Wiki. It utilizes the broad knowledge of a massive number of people on the Internet.

- ✓ *Yahoo's Flickr [20]*: a popular photo-sharing site and provides a mechanism for users to caption their photos. These captions are being used as alternative text for image searching.

- ✓ *CrowdForge [21]***:** a general purpose framework for micro-task markets that provides support for more complex human computation tasks which require coordination among many individuals, such as writing an article.

- ✓ *InnoCentive [22]*: which allows companies with specific needs to share their challenges and specify awards among scientists dispersed all over the world.

- ✓ *CambrainHouse [23]*: built on crowdsourcing foundations, collects, filters, and develops software ideas coming from the crowd.

B) Mobile Based Crowdsourcing

There are some tasks which should be performed in a certain context in the real world and away from the desktop. Such tasks are characterized by the need to be performed in a specific location only, or require the presence of a certain context to solve [15].

Mobile phones are not only empowering most humans on the planet to connect with each other in real-time communication, but also allow us to coordinately share and analyze information, perform computational tasks and soon phones will enable anyone to conduct real-time, peer-to-peer financial transactions. Mobile devices now function as fundamental tools instrumental to billions of economic livelihoods [24].

Mobile crowdsourcing holds immense potential for emerging markets and most research works scratch only the surface of what could become a very powerful ecosystem [18].

Recently, the interest of developing crowdsourcing applications for mobile platforms has grown. This is due to the fact that mobile technology and devices have attracted a huge number of users all over the world as they have become smarter and more equipped with computational power. Some popular examples of mobile based crowdsourcing applications are:

- ✓ *Txteagle [11]:* a mobile crowdsourcing system that enables people to earn small amounts of money by completing simple tasks such as doing translation, transcription, and filling out surveys by using their mobile phones. It uses text messaging service.

- ✓ *Askus [25]:* a mobile platform for supporting networked actions that allows specifying tasks, which are then matched by the system to specific persons based on profiles.

- ✓ *Fashism [26]:* an online community that uses phones as bridges between the physical and digital world. It provides an easy way for customers to get comments on their fashion style while doing shopping by sending a dressing-room photo to the community and getting votes and comments back from the crowds in real time.

- ✓ *Google Maps [27]:* a crowdsourcing application developed by Google to accumulate users map usage and collaboratively build maps, using a user's phone running it.

- ✓ *Ushahidi [28]:* an open-source platform from Kenya, which allows for crowdsourcing crisis information by letting participants submit information on violence through text messaging using a mobile phone, email, and the Web.

2.2.2 Crowdsourcing Based on the Application

The authors of [14] grouped crowdsourcing applications into four categories based on the area in which the crowdsourcing applications are used for and they are voting system, information sharing system, game, and creative system.

A) Voting Systems

Voting systems deal with tasks that require the opinion of a user or individual to carry out the task or to come up with a solution to a given problem.

Voting tasks require a crowdsourcing worker to select his/her answer from a number of choices. The answer that the majority selected is considered to be correct. Voting can be used

as a tool to evaluate the correctness of an answer from the crowd. Some examples of voting tasks are:

- *Commonsense* - Obviously, humans can explain commonsense knowledge about the world, but computer programs cannot. Many studies focused on collecting commonsense knowledge such as The Verbosity system [30] and the Common Consensus system [31] collect commonsense knowledge that is valuable for commonsense reasoning and enhancing the design of interactive user interfaces.
- *Opinions* **-** Gathering opinions from the crowd can be achieved easily in a crowdsourcing system. Mellebeek*et al.* [29] used the crowdsourcing paradigm to classify Spanish consumer comments. They demonstrated that non-expert MTurk annotations outperformed expert annotations using a variety of classifiers.

- *Relevance evaluation* - Humans have to read through every document in a corpus to determine its relevance to a set of test queries. Alonso *et al.* [32] proposed crowdsourcing for relevance evaluation, so that each crowdsourcing worker performs a small evaluation task.

- *Natural language annotation* - Natural language annotation is a task that is easy for humans but currently difficult for automated processes. Recently, researchers investigated crowdsourcing as a source of non-expert natural language annotation, which is a cheap and quick alternative to expert annotations. Some examples of work in this area are:

  - ✓ Akkaya*et al.* [33] showed that crowdsourcing for subjectivity word sense annotation is reliable.
  - ✓ Gao and Vogel [34] proved that crowdsourcing workers outperformed experts on word alignment tasks in terms of alignment error rate.
  - ✓ Jha*et al.* [35] showed that it is possible to build up an accurate prepositional phrase attachment corpus by crowdsourcing workers.
  - ✓ Parent and Eskenazi [36] demonstrated a way to cluster a task of dictionary definitions in MTurk.

B) Information Sharing Systems

Crowdsourcing applications can help to share information easily among users. Some crowdsourcing systems aim to share various types of information among the crowd. These systems allow the crowd to generate content or information so that it could be accessed by other users.

The advantage of information sharing systems is that, the information is not from one particular source but instead composed from various sources, so the user gets generalized information.

There are a number of information sharing systems that use the concept of crowdsourcing to share and process information. Some examples of the popular ones are:

- ✓ *OpenStreetMap [37]:* is a site where many users, living in different geographical regions, contribute, share, and process their location tracks to make a comprehensive online map.

- ✓ *iStockPhoto [38]*: is a web based company offering huge collections of images uploaded and sold by photographers.

C) Games

Games are one means of entertainment and leisure time activity. By their nature games are fun and interesting to play. Till recent times games were developed for the purpose of only having fun and passing time. Nowadays the potentials of games as being a means to acquire the knowledge and skill of their players have been discovered and many researchers are taking advantage of this potential through crowdsourcing games.

The concept of "Social Game" was pioneered by Ahn, who created games with a purpose [39]. The games produce useful metadata as a by-product. By taking advantage of people's desire to be entertained, problems can be solved efficiently by online game players.

There are vast numbers of games that are developed to act as a crowdsourcing application in order to gather knowledge from the players. Some examples of games with crowdsourcing capability are:

- ✓ *Foldit* [40]: a revolutionary new computer game that allows players to assist in predicting protein structures, an important area of biochemistry that seeks to find cures for diseases, by taking advantage of humans' puzzle-solving intuitions.

- ✓ *The online ESP Game* [41]: was the first human computation system, and it was subsequently adopted as the *Google Image Labeler*. Its objective is to collect labels for images on the Web.

- ✓ *The TagATune system* [42]: provides annotation for sounds and music which can improve audio searches

D) Creative Systems

The role of human in creativity cannot be replaced by any advanced technologies. The creative tasks, such as drawing and coding, can only be done by humans. As a result, some researchers seek for crowdsourcing workers to do some creative tasks to reduce production costs.

Although technology advances rapidly nowadays, humans can innovate creative ideas in a product design process but computers cannot. It has no clue about how to solve a specific problem for developing a new product.

One of the best examples of a creative system is Threadless [43], a well-known web-based t-shirt company that crowdsources the design process for its shirts through an ongoing online competition.

2.3 Issues in Crowdsourcing

Lately, researchers and developers have become interested in crowdsourcing after seeing its various advantages. When developing a crowdsourcing application there are various issues to be considered. The authors in [14] stated that the major issues in developing a crowdsourcing

application are: crowd participation, motivation, algorithms, quality assurance, and cheating detection.

In this section, we will explain the issues and present works of researchers and scholars on these issues.

A) Crowd Participation (Content Generation)

Crowdsourcing applications are nothing without the participation of the crowd. The crowd is the one who is responsible for generating the content which can be information, data, or knowledge. The crowdsourcing applications relay on and make use of this crowd generated content to perform a task or solve a problem. The content that is generated by a crowd can be either an explicit content or implicit content [15].

Explicit content generation describes the process in which a number of web users individually produce content. The content production may be carried out independently or as a part of a coordinated effort. Such collections are sometimes seen as making use of the wisdom of the crowd.

Explicitly generated content requires effort by the user and typical incentives are peer recognition or non-material or material benefits, such as payments or vouchers. In [34] the authors investigated how financial incentives impact performance.

Implicit user-generated content describes content that is generated by implicit human computer interaction [44]. Here users generate content by their actions. What is interesting with regard to implicit user-generated content is that there is no extra effort required for the user in order to contribute this content. Nevertheless there might be a cost associated (e.g., loss of privacy).

In a crowdsourcing system, tasks are distributed to a population of anonymous Internet or mobile users for completion. Understanding the demographics of crowdsourcing workers and examining their behavior attracted significant attentions [14].

Downs *et al.* [45] screened MTurk workers by using two previously pilot tested screening questions. Experimental results showed that those that are professionals, students, and non-

workers seem to be more likely to take the task seriously than financial workers, hourly workers, and other workers.

There are several examples where games are successfully used to create content. Ahn and Dabbish [41] have shown that labeling images can be packed and provided to the users in a playful way. A side effect of playing the game is then the assignment of tags and labels to images. In this approach, the game itself is already the incentive for contributing the content.

Another area in which crowdsourcing can be seen as a means of creating content is flash mob. The idea of a flash mob is that, people use digital technologies and coordinate an action that conveys some sort of information or message. If the action has a clear goal this is then considered as a smart mob. Smart mobs are one way of a crowd to participate in crowdsourcing application and generate content [15].

B) Motivation

Humans have the behavioral tendency to require motivation in order to undertake a certain job or task. The motivation can be monetary or non-monetary reward. Monetary reward is a fee or money that is given to a person for performing a certain job, while non-monetary reward is gained by the worker in the form of mental satisfaction or joy for undertaking the task. Motivation plays a great role in making a crowd to participate and use a crowdsourcing application [47].

In addition to monetary reward, a worker gains credibility when his/her task is accepted by the requester. Sometimes, the task requester is obligated to pay every worker who has fulfilled the task according to the requirements. In some cases, workers are not motivated by rewards, but they work for fun or satisfaction.

Although monetary crowdsourcing incentive is dominant, some crowdsourcing systems do not offer monetary rewards to their workers. In the case of YouTube, attention, measured by the number of downloads, is an important driver of contributions [46].

The impact of the incentives on specified crowdsourcing tasks were studied by various researchers. Some of the works are:

- ✓ Harris [47] found that financial incentives actually encourage quality if the task is designed appropriately in resume review.

- ✓ Silberman*et al.* [48] presented that the importance of money is widely appreciated compared to other motivations, with most respondents reporting they do not do tasks for fun or to kill time but instead for the income.

- ✓ In question-answering sites, Nam *et al.* [49] found altruism, learning, and competency as frequent motivations for top answerers to participate, but that participation is often highly intermittent. Besides, they showed that higher levels of participation correlate with better performance.

- ✓ DiPalantino and Vojnovic [50] discussed essential features of a crowdsourcing system and the precise relationship between incentives and participation in such systems are discussed. They reported that rewards yield logarithmically diminishing returns with respect to participation levels.

Many works showed that user interfaces can affect the behavior of crowdsourcing workers.The major once are:

- ✓ By analyzing the waiting time for the posted tasks on MTurk, Ipeirotis [51] found that workers are limited by the current user interface and complete tasks by picking the tasks available through one of the existing sorting criteria.

- ✓ Grady and Lease [52] investigated human factors involved in designing effective tasks on MTurk for document relevant assessment. They found that many of the same workers completed tasks in multiple batches, compromising the experimental control and likely introducing effects of training or fatigue.

C) Algorithms

An algorithm can help to formalize the design of a crowdsourcing system. An algorithm can model the performance of a crowdsourcing system. Many works have been done regarding this concept, some examples are:

- ✓ Wang *et al.* [53] modeled the completion time as a stochastic process and built a statistical method for predicting the expected time for task completion on MTurk and showed how time-independent variables of posted tasks (e.g., type of the task, price of the Human Interactive Task , day posted, etc.) affect completion time.

- ✓ Ipeirotis*et al.* [54] presented an algorithm for quality management of the labeling process in a crowdsourcing system. The algorithm can generate a scalar score representing the inherent quality of each worker.

- ✓ Carterette and Soboroff [55] presented eight models of possible errors for relevance judgments from crowd and showed how each affects an estimate of average precision.

- ✓ Jain and Parkes [56] surveyed existing game-theoretic models for various human computation designs, and also outlined the research challenges by advancing the game theory to enable better design of human computation systems.

D) Quality Assurance

Quality is a key aspect in analyzing the output of an application. In crowdsourcing applications, the quality of the output greatly depends on the content generated by the crowd.

Having a high quality in crowdsourcing application is mostly challenging and this forces many researchers to concentrate on this issue. Some of the works are:

- ✓ Mason and Watts [44] showed that increased financial incentives increase quantity, but not quality, of work performed by crowdsourcing workers. They stated that it is necessary to derive a set of design principles for tasks on crowdsourcing systems to guarantee the output quality of workers.

- ✓ Feng*et al.* [57] carried out experiments and showed that for the same task turkers answered questions quite differently if they were provided different knowledge which suggests that the quality of the answer also differs.

- ✓ Kittur and Kraut [58] showed that adding more editors to an article improved article quality only when they used appropriate coordination techniques and was harmful when they did not.

E) Cheating Detection (Accuracy of Respondents)

In crowdsourcing applications, as the crowd is distributed, the requester of the task doesn't have the chance to know the capability of each individual that performs the task. So the input of incorrect or inaccurate content by a crowd cannot be avoided. Since the inaccurate contents have a negative impact on the output of the application, in developing a crowd application a great deal of concern must be given to the issue of cheating detection.

Various approaches have been proposed by many scholars and researchers to detect cheating while using a crowdsourcing application by the individual crowds in order to get paid within accurate inputs. Some of the approaches are:

- ✓ Cheat-detection techniques are either based on control questions which are evaluated automatically or rely on manual checking by the requester. Eickhoff and de Vries [59] inspected the commonly observed methods of malicious crowdsourcing workers, such as task-dependent evaluation, interface-dependent evaluation, and audience-dependent evaluation. Task dependent evaluation is evaluation on the work of a crowder based on the strength of the task. The interface dependent evaluation is a cheat detection approach that evaluates crowder's work based on the interface they are using to perform the task. The audience-dependent evaluation is going to be undertaken by the end users that use the work of the crowder.

- ✓ For the crowdsourcing systems that control questions are not applicable and manual re-checking is ineffective, Hirth*et al.* [60] presented two crowd-based approaches to detect cheating workers: a majority decision (MD) and an approach using a control group (CG) to re-checking the main task. The MD is an approach used to cheek cheating workers by evaluating their work against the work done by the majority crowd. While using control group is an approach that evaluates the work of a crowd worker by expert workers.

- ✓ A task is considered to be valid, if the majority of the control group decides the task is correctly done. Experimental results showed that crowd-based cheat-detection mechanisms are cheap, reliable, easy to implement, and applicable to different types of typical crowdsourcing tasks. Almendra and Schwabe [61] proposed the use of crowdsourcing to improve precision and recall of current fraud detection techniques for online auction sites.

- ✓ Dawid and Skeene [62] approached the problem of inferring the correct answer from the responses of multiple error-prone respondents when attempting to infer a patient's history based on potentially biased reports from different clinicians. They introduced expectation maximization (EM) model that simultaneously estimates the bias of these different clinicians as well as the underlying latent variable, in this case the patient's medical record.

Chapter Three: Related Work

In this Chapter we will present crowdsourcing applications and research works that are related to our research work and researches done on information extraction. These works are related to our work because they are mobile based crowdsourcing applications, which are explained in Section 3.1 and works done on extraction of information from short messages or SMS messages, discussed in section 3.2.

3.1 Related Works done on Mobile Crowdsourcing

In this Section, we present three works that are done on mobile based crowdsourcing. They are txteagle, MobileWorks, and mClerk.

A) txteagle

txteagle is a mobile based crowdsourcing system developed by Eagle [11]. It enables people to earn small amounts of money by completing simple tasks on their mobile phone which they get paid for in the form of airtime or mobile money.

The author argues that various tasks, such as translation, can be done easily by a human than a computer. So the author developed a system that takes these tasks from computers and sends them to the various users' (the crowd) mobile phones via SMS. Then the crowd will complete the tasks on their mobile phones and send them back to the system via SMS. The system gathers the performed tasks and then analyzes them.

The system was developed in order to be used as a supplementary income for rural and low income populations. It is currently being launched in Kenya and Rwanda in collaboration with the mobile phone service providers Safaricom and MTN Rwanda [11].

As presented in [11], the tasks that are provided by txteagle are:

- ✓ *Transcription:* they have shown that 5 lines of audio text can be written down by hand and then copied into an SMS in less than 2 minutes. Paying proficient users $3/hour to do this work on their mobile phone drops the cost to 2 cents per line.

- ✓ *Software Localization:* It is impossible for software companies to incorporate different natural languages into their interfaces because no translation service exists currently. txtEagle generates a 'phrase book' of relevant words in every Kenyan language and to date txteagle users in Kenya have translated these words into more than 15 local languages.

- ✓ *Citizen Journalism*: By leveraging existing mobile payment system, txteagle enables anyone with a phone to submit a story for free and have compensation directly transmitted back to the handset.

- ✓ *Search Relevancy:* By augmenting the machine learning search algorithms with human input, it may become possible to make dramatic improvements in relevancy for specific queries.

- ✓ *Surveys & Market Research:* the author stated that in countries such as Kenya, where even the most isolated areas generally have GSM reception, it becomes possible to remotely administer surveys using text messaging. Phone-based surveys are an efficient and economical method of collecting a wide variety of information including market research, census, health, activity, and commerce data.

The txteagle system is an active system compared to other crowdsourcing systems, such as Mtrunk, in the sense that the system selects the most appropriate tasks for a txteagle user by customizing the task difficulty for a given user. This is achieved by enabling the system to learn about the user's areas of expertise the more the user responds. This helps the system improve the task assignments and helps the user to efficiently generate high-confidence task responses in the shortest amount of time.

The researcher in this work has come across two issues while designing this mobile crowdsourcing platform: ensuring tasks have been completed correctly and assessing the accuracy of an individual user**.**

The author tackled the first issue by sending out the same task repeatedly and verify that they are getting the same response across multiple, independent users. It is noted that each response *yij* comes from an error-prone user *j* that has an accuracy *pj* drawn from a known distribution of accuracies and initially assume that *pj* is independent of a particular task *xi* and

true answer *yTi*, such that *Pr*(*yij*= *yTi*| *xi*) = *Pr*(*yij*= *yTi*) = *pj*. The author assesses the validity of a response through majority voting to infer a single integrated response y that is simply the most popular response from a set of users. The probability this response is correct is noted as *qi* = *Pr*(y*i*=*yTi*) and is referred to as the integrated quality of the most popular response.

The second issue was handled by applying Dawid and Skene's expectation-maximization (EM) model. Dawid and Skene [62] demonstrated that maximum likelihood estimates of user error rates can be calculated with expectation-maximization (EM) with noisy responses from a set of error-prone individuals. Based on this model the researcher of txteagle has developed a similar model to infer correct answers, estimate txteagle users' accuracy levels, and infer their expertise (individual accuracy conditioned on task type) based on their response history.

B) MobileWorks

MobileWorks is a mobile phone-based crowdsourcing platform developed by Narula *et al.* [64]. It provides character recognition (OCR) tasks that can be completed by workers on low-end mobile phones through a web browser. This makes MobileWorks a mobile-web based crowdsourcing platform.

MobileWorks consists of three main components, the preprocessing stage component, user interface component, and the post processing component.

The first component takes scanned paper documents and generates many small OCR tasks for each document. Because of the limited screen size of mobile phones, documents have to be chopped into small pieces of one or two words.

The second component, the user interface, enables workers to perform OCR tasks. The interface is minimal, the user logs in using his/her username and password. The user can start doing tasks immediately after login.

The third component, the post-processing stage, reassembles pieces of completed OCR tasks into a digitized copy of the original document and ensures quality. After the different workers digitize the pieces using the MobileWorks web application and submit it to the server, these smaller pieces are then put together to create a digitized copy of the document.

After conducting a pilot study, the researchers evaluated this system from the point of view of quality, accuracy, and efficiency.

They maintained quality by using multiple entries. Each task is distributed to two workers until two of the answers match. If a worker provides an incorrect answer, his/her quality rating decreases. Conversely, a worker that provides a correct answer will see an increase in quality score.

For the accuracy, they found that the overall accuracy of the workers, without considering multiple entry error detection, was about 89%. Though they did not explicitly test the accuracy using multiple entries in the experiment, in cases where errors were independently distributed between workers, dual entry accuracy would be 98.79%, and for triple entry 99.89%. According to them, this suggests that by using multiple entries, it is possible to provide very high quality crowdsourced human OCR work on simple mobile phones.

In terms of efficiency, using MobileWorks, participants were able to complete 120 tasks per hour. In addition, improvements in wireless infrastructure will improve data transfer rates, decreasing the time it takes to complete each task.

C) mClerk

mClerk is a new platform for mobile crowdsourcing in developing regions. mClerk sends and receives tasks via SMS, making it accessible to anyone with a low-end mobile phone. It was developed by Gupta *et al.* [63].

mClerk is not limited to only sending text messages. It leverages a protocol to send small images via ordinary SMS, enabling novel distribution of graphical tasks. These protocols are Nokia's Smart Messaging (SM) and Ericsson's EMS (Ericsson Messaging Service). They are device dependent protocols that support sending binary picture messages via SMS with restrictions. The pictures to be sent are restricted to 74x28 pixels for SM and 64x16 pixels for EMS. The SM picture message is actually three concatenated binary SMS messages.

The mClerk system starts with a scan of a paper document, segments it into word images, sends each image via SMS to users' phones, receives back the users' responses, probabilistically verifies them, pays the users and aggregates responses into a digital document.

It has four modules: image segmentation, a mobile crowdsourcing platform, word aggregation code, and a payment mechanism.

The system segments scanned pages of handwritten or printed text into separate words, considering various issues such as multiple ink and background shades, paper skew, text skew and salt and pepper noise. The segmented word images are then binarized and resized. The segmented word images are sent to the users' mobile phones, where users are expected to key in each word and send back the text.

The verification of responses in mClerk is typically done by comparing multiple responses for the same task. Since the users are sending back transliterations, there could be a large number of cases where similar but distinct transliterations point to the same word. They modified their algorithm so that two responses are considered equivalent if both of them transliterate back to the same word in the local language, where two equivalent responses for the same word are considered to be correct. If the first two responses to the same word do not match, then the system sends the word to a third person and continues this process until it gets two equivalent responses.

Compensation is in the form of mobile airtime. They send a leaderboard message at the end of the day listing the names and earnings of the day's top 5 leaders to motivate the workers.

The researchers evaluated the system by performing an experiment and calculated the accuracy. Accuracy was defined as the number of words correctly digitized by the system divided by the total number of words. The system achieved an accuracy of 90.1%, which can be considered high given the intermediary transliteration.

3.2 Related Works on Information from Short Messages

As our research work mainly involves extracting information from short messages, in this Section, we have reviewed and presented works that have been done on extraction of information from short messages or SMS messages. There are two research works discussed in this section.

A) Extracting Information from Short Messages

Cooper *et al.* [65] came up with an information extraction component that uses pattern matching approach to extract data from short messages, i.e., E-mail or SMS messages.

What they propose is that, words in a message can be portioned into the syntactic structure, terms from the domain of discourse and the data being transmitted. So by using pattern matching approach it is possible to separate the three aspects as the structure is supplied as a template; domain terms are the metadata of a data source (or their synonyms), and data is extracted as those words matching placeholders in the templates.

Their proposed system is a semi- automated system in which the information passed by e-mail or SMS message will be stored in a database, so that the data will be extracted using pre-prepared structural template and metadata.

To extract information for storage, they used the metadata they pre-prepared as a basis for the collection of terms from the messages, augmented with synonyms to cope with equivalent terms. Then the terms are combined with the templates to generate patterns for matching.

When a match is found, the data is extracted from the parts of the text matching data placeholders and the result is turned into appropriate database updates. The matching process uses a maintained context to deal with anaphoric references and the update generation creates a mixture of entity creation and property update commands. So, the proposed system comprises three phases:

- ✓ the generation of a collection of sentence structure patterns,
- ✓ the use of the collected patterns to locate new data from the message, and

- ✓ The use of that data to derive database update statements to store the information found.

Due to the fact that the authors did not say anything about testing the system and its outcome in the paper we referred, we can't report the success of the proposed system.

B) Information Extraction from Short Text Message in Bahasa Indonesia for Electronics

Muhammad and Hendratmo [66] proposed an information extraction from SMS text messages using pattern matching method with the hypothesis that short text messages can be extracted using some pattern. In order to reach this hypothesis they studied the characteristics of short messages and their technical effect on the information extraction process and found the following characters in SMS messages:

- ✓ Usage of slang words
- ✓ Sentences do not follow word sequence (subject, verb, noun, etc.) as in the right grammar
- ✓ Usage of abbreviations of the intended words
- ✓ Random-cased characters writing
- ✓ There is a convention(s) pattern used between text messages from same domain.

Using this hypothesis, what they proposed was a method for observation of inter-slot pattern. Inter-slot pattern is observed as a mean and standard deviation. i.e; pattern is constructed from characteristics of the slot that will be extracted and the pattern is realized in the form of dictionary matching and the pattern of the filler. If the distance of slot filler from another slot filler (or keyword token) is consistent, then that information can be used to locate particular slot filler relative to some keywords or other slot filler that is already discovered. Figure 3.1 shows the proposed model. The steps involved in the proposed system are:

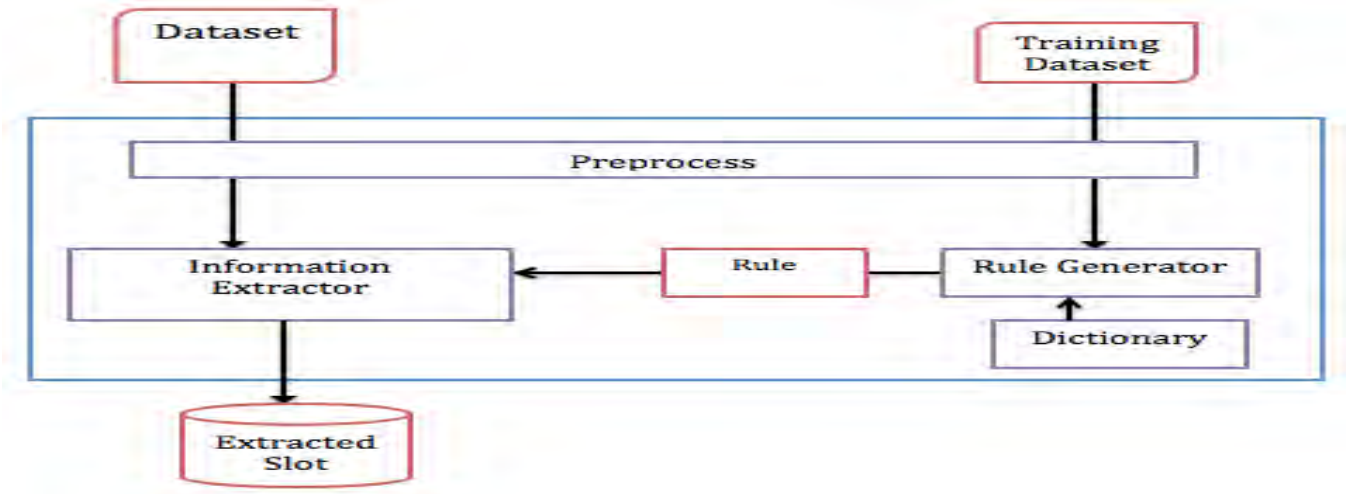

Figure 3.1: Proposed Information Extraction Model [66]

- ✓ The dataset is divided into training and testing and pre-processed before treated further. Preprocessing aims to transform input text in order to be understood by the system.
- ✓ Manual labeling on dataset: A label is given to a slot containing information that should be extracted to that slot. In order to use inter-slot pattern identification approach, keyword labeling is also needed to be done.
- ✓ Rule generator function to generate rule based on particular patterns. From labeled dataset, analysis is conducted to determine which method to use to generate the rule. If there are more than one pattern resulted then they can be eliminated or combined to form a final rule.
- ✓ Dictionary is used to match tokens in a specific domain.
- ✓ Output rules are used to extract information from dataset. Rules are implemented on regular expression form.
- ✓ Information Extractor is the part to execute the rules on the input dataset. Information from input dataset will be extracted based on the rule used to fill target slots.

After performing test experiment on the system they developed using the proposed inter-slot pattern identification approach to extract the information from the short messages, the experiment gives a result of 76.5% precision and 70.7% recall. According to the authors the results indicate that patterns on the rules are relevant enough but not all data entries match with the patterns. Some errors are also caused by syntactic nature of short text messages.

3.3 Summary

The above mentioned crowdsourcing works all have one thing in common; they accept words as an input from the crowd and analyze the word. But they all accept the word and only do word level segmentation and don't try to analyze the semantic but our proposed system is expected to perform semantic analysis on the text that is sent from the crowd by using ontology based text extraction and analysis.

Our system extracts four basic set of information from a crowder's incoming text: product, price, place and date, which are the needed information to analyze the price of an agricultural product.

Concerning the other development issues such as motivation, like the above mobile based crowdsourcing works, our system is going to provide air time payment as a monetary incentive. But unlike the others, we are going to use the air time payment and the history of the crowder's participation and accuracy to analyze the quality of the crowder's information.

To check the accuracy of the crowder's information, we are going to use the approach of Majority Decision (MD) by applying Dawid and Skene's expectation-maximization (EM) model [62].

The reason why we choose ontology based information extraction is due to the fact that, ontology based information extraction can easily describe the semantic characteristic of a concept compared to a pattern matching approach and this makes it easy to extract information from a text message. Ontology based extraction can help us in identifying each word's meaning related to agro market concepts, like product and price, from the crowd SMS messages.

Chapter Four: Design of Mobile Based Crowdsourcing for Agricultural Market Information

In this Chapter, we present and describe the proposed architecture of the mobile based crowdsourcing for agricultural market information system that is implemented in mobile phone and a server. The system is designed to work in any cell phone that supports J2ME (Java 2 Micro Edition), Connected Limited Device Configuration (CLDC) 1.0 or 1.1 and Mobile Information Device Profile (MIDP) 2.0 or 2.1 that has an existing SMS messaging service.

Even though the proposed system is a client server based system, as the client is only involved in sending and Receiving SMS message which can be simulated using Java ME (Java Micro Edition) Wireless Messaging API (Application Programming Interface), we focus on the five basic components involved in the server side of the proposed system.

4.1 Introduction

Mobile based crowdsourcing for agricultural market information system, as the name refers, is a mobile based crowdsourcing system, which is expected to analyze and deliver generalized agricultural market information based on the market information provided by the crowd. The information can be sent by anyone who wants to share their knowledge of the current agricultural product market price, which means the system is open to anyone. The crowd is going to provide this information via the already existing SMS messaging service on their mobile phones. The system is designed to handle massages written in English or Amharic language.

Our proposed system architecture is designed with the conscience that any phone has the capability to send SMS message and the crowd can use the existing SMS service, provided by Ethio-Telecom, to send the messages to the server.

The aim of the proposed system is to automatically extract, analyze and store the four basic agricultural market information, which are: the name of the agricultural product, the price of the product, the location of the market the product is available at and the day on which the product is sold at that price, from the incoming crowd SMS messages. And finally deliver the stored information when a request comes from a user via SMS message.

The system, in the first place, needs to accept SMS messages that contain agricultural market information from the crowd. After storing the messages, the system is supposed to analyze the content of each message and aggregate the information from the crowd to output generalized market information. Due to the fact that the crowd is going to use natural language to write the SMS message, it will be difficult for the system to analyze the content of the message. Due to this fact, before analyzing and aggregating the information, the system has to perform information extraction on each message to get the required piece of information.

In the proposed system, there are five basic activities that are done by the server. The first one is to accept the incoming messages from the crowd and place them in a database. The second activity is to analyze the content of each stored message and extract the required agro-market information. As we are adopting the ontology based information extraction approach, the extraction will be done with the help of a pre-prepared ontology. The third activity uses the extracted information and aggregates the market information of a given product sold at a particular market place on a given day and stores this agro-market knowledge in a database. Managing the crowd is the fourth activity that is done by the server. This activity involves storing and analyzing the information of the crowd that participated in sending the SMS message. Once the extracted agro-market information is analyzed and stored in the database and if a user requests to get that agro-market data the fifth activity, that processes the request and retrieves the required data, will be carried out by the server.

4.2 The Proposed System Architecture

The proposed architecture for mobile based crowdsourcing for agricultural market information system consists of five major components that carry out the five activities done by the server. These are: SMS manager, Information Extractor, crowd data analyzer, crowd manager and request manager. Table 4.1 lists the main components of the proposed system along with their modules and describes their purpose and Figure 4.1 illustrates the generic architecture of the proposed mobile based crowdsourcing for agricultural market information system.

Table 4.1: Components of the Proposed System

Component	**Modules involved**	**Purpose**
SMS Manager	SMS Manager	responsible for managing the incoming and outgoing SMS to and from the server
Information Extractor	Pre-linguistic processing	Required for analyzing the content of each message and extracting the four basic agricultural market information from the SMS messages
	Language identifier	
	Ontology based IE	
Crowd Data Analyzer	Knowledge classification	accepts the extracted information and aggregates the market information for a given product
	Ranking	
Crowd Manager	Crowd Manager	responsible for managing the crowd participation and motivation
Request Manager	Request identification	accepts and processes requests for market information from the system users via SMS messages
	Query generator	
	Data retrieval	

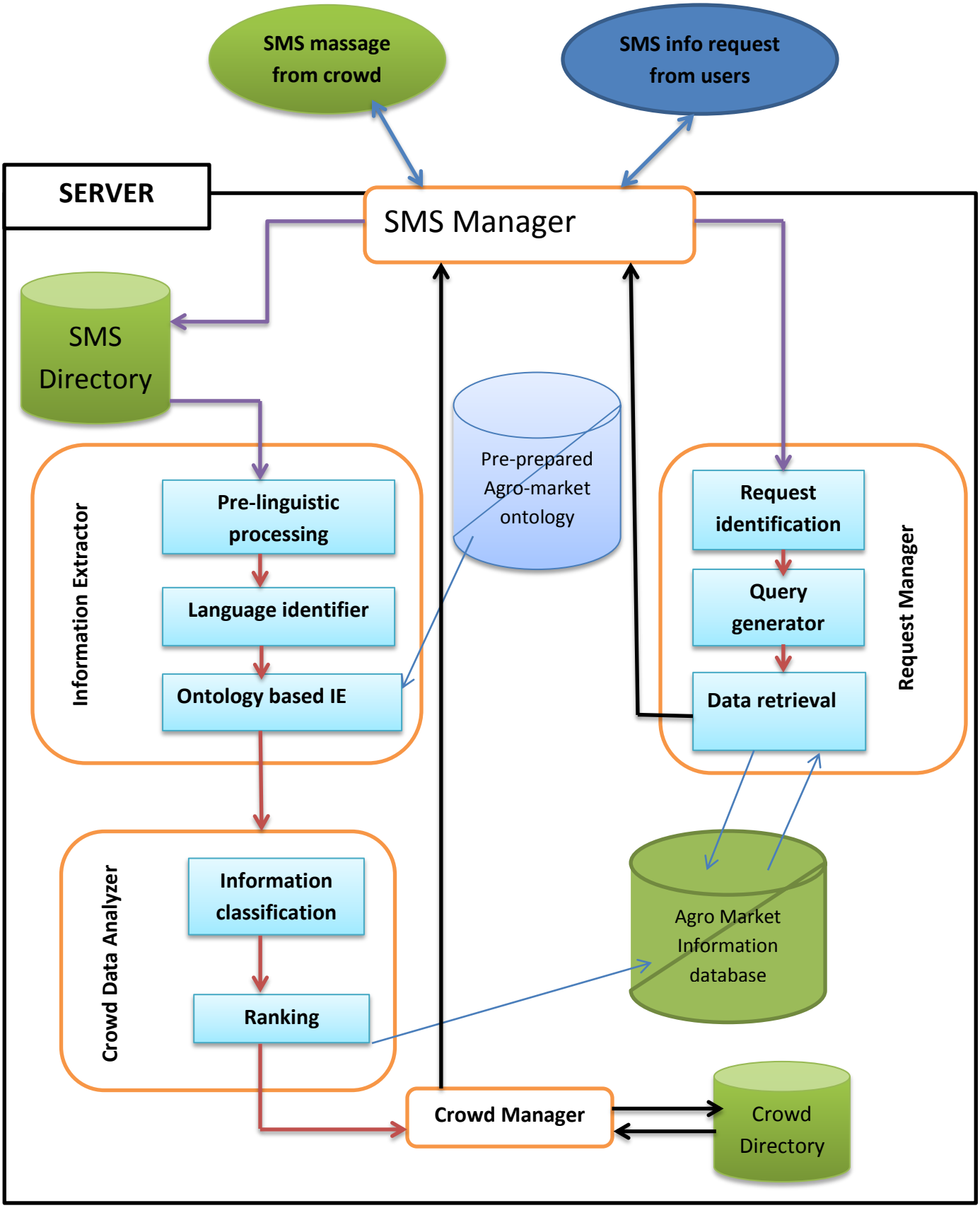

Figure 4.1: Architecture for Mobile Based Crowdsourcing for Agro Market Information System

4.2.1 SMS Manager

When a crowder sends an SMS message from his/her phones using the SMS services on the phone, accepting the SMS message from the crowd is the initial task of the system server. The SMS manager component is responsible for this task. This module is the listening end point for the server, listening to incoming and outgoing SMS messages. It acts as a bridge between the crowder and the server. It is implemented as a servlet class and handles the connection to the SMS directory database.

When a crowder sends an SMS message to the system, the SMS manager will accept the message and stores it in the SMS directory database. After the message is processed and when the crowd manager component wants to send a notification for the crowder, the SMS manager will accept the notification and sends it to the crowder's mobile phone.

The SMS manager is also responsible for accepting an SMS request for market information that comes from the end user. It will pass the request to the request manager component and once the request is processed it accepts the result in the form of SMS message and sends it to the requester's mobile phone.

As the SMS manager is responsible for accepting two types of SMS messages, crowd SMS and request SMS message, it differentiates them by inspecting the beginning of the SMS message. If the SMS message starts with #req#,the SMS manager identify it as a request message and pass it to the Request Manager component but if the message doesn't start with #req# it will identify it as a crowd SMS message and store it in the SMS directory database as shown in Algorithm 4.1.

Algorithm 4.1: SMS Manager

```
1. INPUT
   Incoming SMS message
2. CHECK for #req# on beginning of message
      a. If not found THEN insert message onto SMS directory database
      b. If found THEN call Request Manager component and pass message
      c. END IF
3. SEND notification
END
```

4.2.2 Information Extractor

As text messages are written in natural languages, which are understandable by humans but useless for computer systems, it is almost impossible for a computer system to use them directly without performing some kind of natural language analysis. In our case, our system requires information extraction to be done on the SMS messages to automatically extract the basic agricultural market information from the SMS message. So, this task is performed by the information Extractor component of the server.

The Information Extractor will fetch a group of SMS messages found in the SMS directory database every six hours and performs information extraction on every message to extract the four basic market information (product, price, market place and date), which are explained earlier, and finally passes the extracted values to the Crowd Data Analyzer component. The reason why we choose six hours is by assuming that there will be a large number of SMS message in the database that the system could be working on and the another reason is, as the agro market is dynamic and changing every time and we want to show this property in our system by updating the information provided by the system every six hours.

Information extraction can be done on a text using two approaches, the NLP approach and the ontology based approach. Because we are working on a domain based text we choose to adopt the later approach for our system. Therefore, the information extractor component will use pre-prepared agro-market OWL ontology for the extraction process.

The ontology which is used by this component is a static ontology, which we pre-prepared using OWL ontology by studying and modeling the concepts involved in agro market information exchange.

The Agro market ontology consists of the four basic concepts in agro market information, product, price, place and date. The four concepts are the classes in the ontology and the classes contain various instances as shown in Annex C and also contain relationships among the classes, such as “PRODUCT has a PRICE” and “PRODUCT sold at PLACE”.

This component uses the ontology to identify and extract words that have agro market concept from the incoming crowd SMS message by matching each word in the SMS message with the instances of the classes in the ontology.

The component has three basic modules as depicted in Figure 4.2. They are language Identification module, pre linguistic processing module and the ontology based information extraction module.

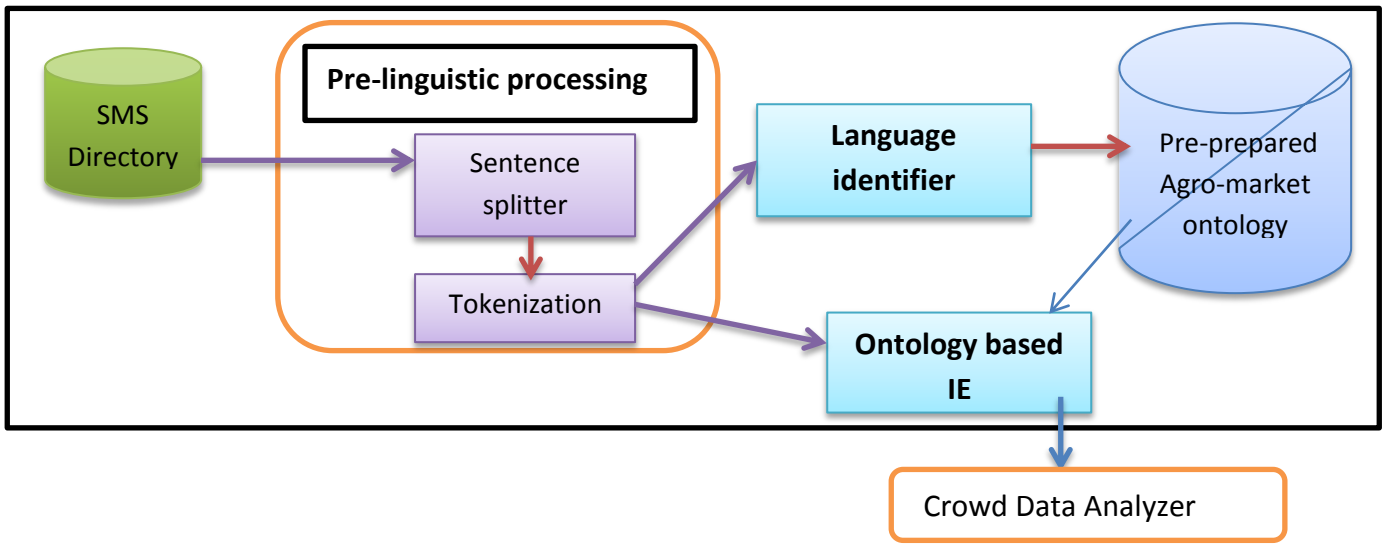

Figure 4.2: Information Extractor Component

Pre-linguistic Processing Module

The first task to be carried out whenever a system processes natural languages is to perform pre-processing on the text or document. The pre-linguistic module performs pre-processing on each SMS message when the SMS text is fetched from the database. The pre-linguistic model carries out two basic tasks. They are:

- **Sentence splitter**: if the message has more than one sentence, this module will split the message into the number of sentence and treat them separately. This is done by the Split function of the String Java class.
- **Tokenization**: the module will separate the message into 1-gram and 2-gram words, so it could be used by the information extraction module. This is done by the StringTokenizer function of the String Java class.

Language Identifier Module

Because we are dealing with SMS messages written in Amharic and English language, the system has to know in which language the message is written. In order for the information extraction module to perform extraction, the language identification module has to identify the language in which the message is written.

Once the SMS message is tokenized, this module evaluates the Unicode values of each character of every word in the SMS message and matches the values with Amharic and English Unicode values to check if the characters are Amharic or English ones. After this the module calculates the percentage of the presence of Amharic or English characters in the message and if all of the characters are Amharic then the language of the message will be taken as Amharic and if all the characters are English the language will be identified as English. Messages with mixed characters or characters other than Amharic or English will be ignored by the system.

If the message is written in Amharic, the module will fetch the Amharic pre-prepared ontology and if it is written in English it will fetch the English ontology to be used by the information extraction module. If the message is written in other language, other than the two languages character list, the text will be deleted from the SMS directory.

Ontology Based Information Extraction Module

The purpose of the information extractor component is to extract the four basic agro market information from the received SMS messages. The information extraction module uses the tokenized SMS messages from the pre-linguistic processing module and the selected ontology by the language identification module as an input to extract the required information.

The module will implement ontology matching algorithm to extract the information. After performing the extraction the output is stored in a temporary Database which will be an input for the Crowd Data Analysis component of the system.

After accepting the tokenized message from the pre-linguistic processing module and using the selected ontology by the language identification module, this module parses all the words in the message and tries to see if the words exist in the ontology definition and identifies the class to which they belong to. If a word exists the module then queries the definition and property of the word from the ontology in order to extract the word as being a product, a price, a place or a date. Then the extracted words will be stored in the temporary database. The algorithm is given in Algorithm 4.2.

Algorithm 4.2: Information Extractor

1. BEGIN
2. REPEAT
    a. READ message from SMS directory database
    b. IF message's number of sentence > 1 Sentence THEN
        i. Split message into message's number of sentence using whit space
        ii. Tokenize words of each Sentence
        iii. Call Language Identifier component and pass Tokenized word
    c. ELSE
        i. Tokenize words of each Sentence
        ii. Call Language Identifier component and pass Tokenized word
    d. END IF
    e. BEGIN
    f. REPEAT
        i. CHECK Unicode value of latter = Amharic Unicode value
        ii. COUNT the number of latters with Amharic Unicode value

g. UNTIL end of Tokenized word
    h. IF COUNT== the number of latters in Tokenized word
        i. LABEL message language as AMHARIC
        ii. INPUT AMHARIC ontology
        iii. SEARCH Tokenized word in AMHARIC ontology
        i. IF found THEN RETRIVE Tokenized word meaning from AMHARIC ontology and PASS it to Crowd Data Analyzer
        iv. IF not found THEN ignore.
    i. ELSE
        i. LABEL message language as ENGLISH
        ii. INPUT ENGLISH ontology
        iii. SEARCH Tokenized word in ENGLISH ontology
        ii. IF found THEN RETRIVE Tokenized word meaning from ENGLISH ontology and PASS it to Crowd Data Analyzer
        iii. IF not found THEN ignore.
3. UNTIL end of SMS directory database

END

4.2.3 Crowd Data Analyzer

Our system is based on the idea that a crowd knows best. In other words if most of the crowd agree on something then, we can take it as being true or correct. When we take a look at agricultural product prices they vary from place to place or from time to time and this makes it difficult to come up with a generalized market price. But one thing that can be observed from any market place is that, the price of a particular agricultural product at a given market place and its vicinities is relatively similar or is in a certain range.

So what we propose is if the majority of the crowd agreed on a certain range of price for a given agro product at a given market area on a particular day, then we can take the average of the range price as the generalized price for the product at that market area. This will be implemented in our system by the Crowd Data Analyzer component.

The component uses data stored in the temporary database, which is generated by the Information Extractor component, and analyzes the four basic agro-market information and comes up with generalized agro market price information which will be stored in Agro-Market knowledge database so that it could be easy for decision making or answering a request for agro market information.

In order to aggregate and generalize the price of a given product, this component first groups and selects message entries with the same product sold at the same market place and computes the average price of the product by applying EM (expectation maximization) algorithm and finally storing the aggregated data in a database as product, price, market place and date column values.

The Crowd Data Analyzer has two modules as shown in Figure 4.3. They are: knowledge classification module and ranking module.

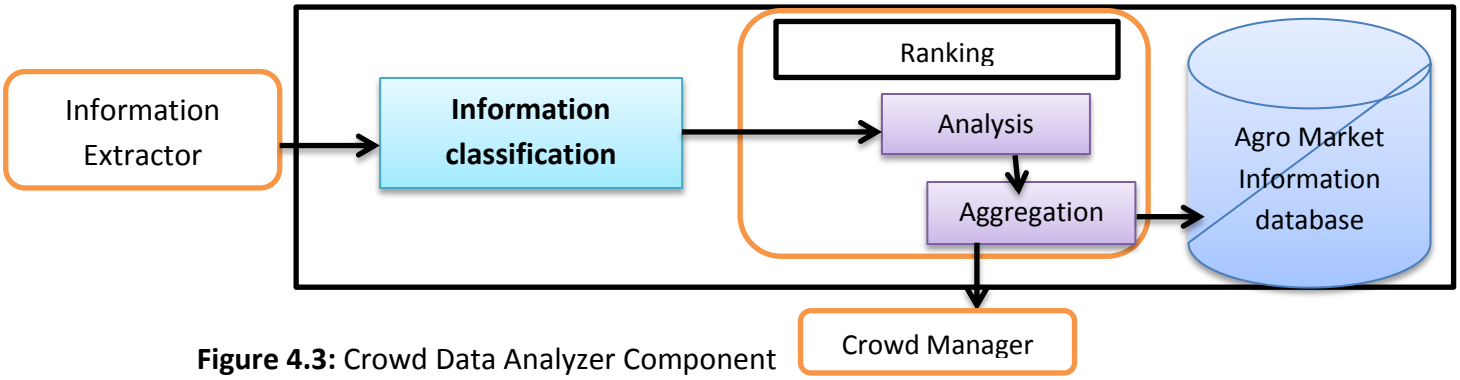

Figure 4.3: Crowd Data Analyzer Component

Information Classification Module

This module is needed for classifying the various extracted data by the Information Extractor component, based on the type of product, market place and date of the SMS message. This will retrieve and group SMS messages that mention the same product and same market together by creating a query that selects the entries from the temporary database so that they will be aggregated by the ranking module.

Ranking Module

In a crowd based system the main goal is to find out or analyze what the majority of the crowd thinks, believes or says about a particular thing. In our system what we want to get from the crowd is the price of a given agricultural product. So, we have to analyze the price range that the majority suggests for a particular product.

In the ranking module, the major task is to accept the classified information and perform statistical analysis in order to find the majority accepted range of price and calculate the average price of the product. The prices that don't fall in the calculated average price will be conceded as outliers.

The module will apply the EM (expectation maximization) algorithm to find the majority price range. This algorithm uses the approach of majority decision, which tries to find the value which is suggested by the majority crowd as shown in Algorithm 4.3.

As depicted in Figure 4.3, the final output of the analyzed and aggregated information is stored in the agro market Information database.

Algorithm 4.3: Crowd Data Analyzer Ranking

1. INPUT
   Extracted data by the by the Information Extractor component
2. SELECT all PRICE Value from product table
3. CALCULATE average price value
4. REPEAT
   i. Compare each product price with average price value
   ii. IF difference margin is less than 5 values THEN RANK product price as CORRECT PRICE
   iii. IF difference margin is greater than 5 values THEN RANK product price as INCORRECT PRICE
5. UNTIL last row of Product table

END

4.2.4 Crowd Manager

The focus area in crowdsourcing research is crowd management. The issues that are raised in this area are motivation for the crowd participation, quality of the content generated by the crowd and accuracy of responses by the crowd. Our research focuses on the motivational aspect and accuracy of the response (cheating detection).

What we are proposing is that in order to motivate the crowd and make them keep on providing the required information we have to reward the crowders, who provide information that agrees with the majority of the crowd in the form of phone air time every time they send five correct SMS messages.

For the case of accuracy of the information we are proposing a trust based approach. In this approach the system will keep track of SMS messages sent by every crowder and when they give correct information their trust level will increase and when they send incorrect information their trust level will be decreased.

Once the Crowd Data Analyzer finishes the aggregation process and identifies the outlier messages, the crowd management component sets flags on each message sender, to increase and decrease their trust level, and updates the Crowd Directory. It then sends notification to the SMS Manager component, which then sends the notification or reward to the crowder's phone.

4.2.5 Request Manager

When the system is done processing the incoming crowd SMS messages it produces an Agro-market information database that has the four basic agro market information. This information can be utilized by any decision making body or any interested individual by sending a request SMS message to the system.

When a request SMS message comes, the SMS Manager will accept it and will identify it as a request SMS message as it will have #req# in the beginning of the message, we used this to distinguish request messages from the crowd messages, so it will pass the SMS message to the Request Manager and after the request is processed, the SMS Manager accepts the result and sends it to the requester's phone.

The Request Manager, after accepting the Request SMS message, identifies the product and market place from the request and then generates a query that specifies the request and sends a query to the agro-market information database. Once the data is retrieved it will then be converted into an SMS message and will be sent to the SMS Manager.

The Request Manager has three modules as shown in Figure 4.4. They are: request identification module, query generator module, and data retrieval module.

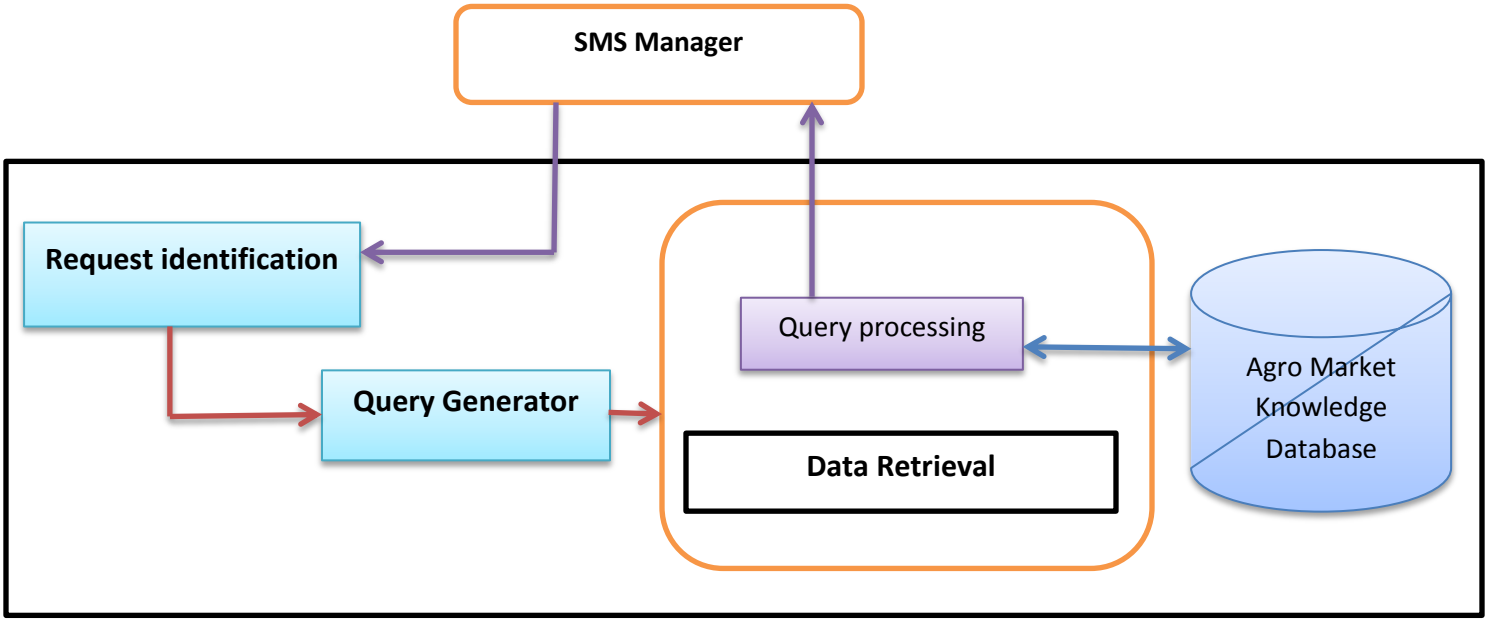

Figure 4.4: Request Manager Component

Request Identification Module

In order to process the request, the system has to identify which product and what market the requester wants to have the information about. This module automatically identifies them using Java string manipulation methods from the message by taking the word next to the #req# string as a product and the word after that will be taken as the place.

Query Generator Module

After the required information is identified, this module will generate MySql query that can access the information in the Agro-market knowledge Database. The generated query will be sent to the Data Retrieval module.

Data Retrieval Module

This module is responsible for accepting the generated query and sends it to the Agro-market Information database and when the response to the query comes in, it converts it to an SMS message something like *"The product X is being sold at y market Today"*. It then sends the message to the SMS Manager, which finally sends the message to the requester's phone.

4.3 Summary

Our proposed mobile based crowdsourcing system will provide an aggregated and summarized market price information to any user by accepting an SMS message from the crowd which provides an agro market information and then analyzing and aggregating the content of massage by going through all the five components of the proposed system, described earlier in the proposed system architecture. The output of the proposed system is an aggregated agro market price range for a particular agricultural product.

Chapter Five: Implementation

In the previous Chapter we presented the detail of the proposed mobile based crowdsourcing for agricultural market information system architecture. This Chapter discusses the prototype implementation of the architecture by presenting the objectives of the prototype, tools and technologies used to develop the prototype and the implementation details.

5.1 Overview

The prototype developed in this thesis aimed at providing proof of the concepts presented in the detailed explanation of the proposed mobile based crowdsourcing for agricultural market information system architecture. To achieve this principal goal, the components in the architecture have been implemented as per the theoretical specifications formulated for each component.

We have developed a prototype which can enable a user (crowd) to send an SMS message that contains agro-market information, written in either Amharic or English language, to the system server. The server after accepting the messages performs information extraction and data analysis on the messages to produce generalized and aggregated agro-market information.

The prototype has two parts, the mobile component part and the server component part. In the mobile component part we simulate the sending of an SMS message from a mobile to a server. In the server component part, we have implemented the five components of the server that are described in the proposed system architecture in the previous chapter, which are SMS Manager, Information Extractor, Crowd Data Analyzer, Crowd Manager and Request Manager.

5.2 Development Environment

The prototype is written with the Java programming language to take the language's advantages of machine independency and its ease and availability of mobile programming. The prototype is developed and tested in a system with Intel Core i3 CPU of 2.1 GHZ speed, a 2 GB RAM, and a Windows 7 operating system.

Several tools and technologies were utilized for the purpose of developing the prototype. The following is a list of programming, communication, database management, and ontology development and reasoning environments used in the prototype implementation.

- NetBeans Integrated Development Environment (IDE) version 7.3.1 is used for developing the prototype. This version of NetBeans uses Java 2 Standard Edition (J2SE) specification version 7 with Java Development Kit (JDK) version 7.

- Java Micro Edition (Java ME) Software Development Kit (SDK) version 3.3 for CLDC (Connected Limited Device Configuration) of the MIDP (Mobile Information Device Profile) is used for developing the mobile component of the prototype.

- sun_java_wireless_toolkit version 2.5.2 emulator for Java, which is used as a mobile device emulator.

- MySQL database server version 5.7.71 is used for persistent data management (for storing the incoming SMS messages, the crowd information and the Agro-market knowledge data) on the server.

- Java Servlet application program interface (API) version 3.0 is used for bridging the mobile version and the server side of the prototype.

- Apache Tomcat version 7.0.50 is used as a web container for Java Servlets.

- Protege 4 OWL Editor Version 4.0.115 is used for developing the agro-market ontology.

- Apache-Jena application program interface (API) version 2.11.0 is used for implementing the ontology and reasoning on the ontology.

5.3 Implementation Details

In this section, we will look into the details of the implementation of the prototype for the proposed system. The Java classes which implement the algorithms, in Chapter 4, for the basic components of the proposed system along with their tasks are presented and in addition Annex B shows sample code of the prototype.

5.3.1 Mobile Component Part

In developing the prototype for the proposed system, we have to simulate the SMS message being sent from the crowd in order to prove the validity of our theoretical system design. Using Java ME SDK we created a MIDlet application that uses the Mobile Information Device Profile (MIDP) of the Connected Limited Device Configuration (CLDC) for the Java ME environment, which works on mobile phone simulator that can compose and send an Amharic or English SMS message to the server by using HTTP request.

For simulating the real world scenario of sending an SMS message to a server, we require two mobile phone emulators. One emulator will be acting as the crowd mobile phone and will be used for sending the SMS message and the other one will be used for receiving the sent SMS message and acts as an SMS gateway that delivers the SMS message to the server using HTTP connection as shown in Figure 5.1.

For developing the Mobile component part we have used **sun_java_wireless_toolkit** version 2.5.2 emulators. In order for the emulators to support Amharic characters we have modified the emulator property of the Gray phone emulator, found in the **sun_java_wireless_toolkit** installation directory, to support Nyala font.

In the prototype we have implemented the SMS messaging MIDlet application, in two Java ME mobile application classes called **Messaging()** and **SmsSenderThread()** class. The **Messaging()**class contains java ME code that sets the MIDlet interface for the sending and receiving emulators and also contains the code that connects the receiver emulator to the server with an HTTP connection. The connection is achieved by calling an HttpServlet that resides in the server. The HTTP servlet is implemented in the prototype by a Java servlet class named **SMSMangerServlate()**which runs on the server acting as a port listening to incoming SMS messages from the receiver emulator.

Figure 5.1: Screen shot of two Emulators Sending and Receiving SMS Message

5.3.2 Server Component Part

The server component part is the part which implements the proposed system server components. As mentioned in chapter four the proposed system server will have five basic components. This section explains in detail how the theoretically explained server components

are implemented in the prototype by presenting the classes involved in the development of the components found on the server part.

SMS Manager Component

It is the server component that is responsible for accepting an SMS message from the crowd by providing an HTTP connection port for the incoming message. We have used a dynamic port for the server. The SMS manager component is implemented in the server by a servlet java class, SMSMangerServlate(). The class accepts an incoming message from the emulator using the DoGet() method of the java servlet class and finally stores it in the SMS directory database.

Information Extractor Component

Information extractor component is implemented by **InfoExtractor(),** a Java class that undertakes the pre-linguistic, language identifier, and ontology based IE modules of this component. The **InfoExtractor()**java classcalls twojava methods named **Toknizer()**, **LangIdentfier()**, and **OntoIE()**java class that undertakes the three modules of the component.

The **Toknizer()** function accepts each massage and applies string split() and string tokenization() methods of the Java string class. To tokenize each word in the message white space is used as the delimenator character. Finally the tokenized words will be stored in a string array.

The **LangIdentfier()** function accepts the tokenized string array and evaluates the Unicode value of each character in the word using GetUnicode() method and finally computes the percentage of the whole string characters by counting the Unicode values that fell in the Amharic character Unicode value and the English character Unicode value. If all of the Unicode values fell in the Amharic character Unicode value range then the language of the message will be taken as Amharic and if all of the values fell in the English character range then the language will be identified as English.

The**OntoIE()**java class accepts the tokenized string array and the language type from the **InfoExtractor()** class. The class executes ontology reasoning on each word in the string array to check if the word is found in the pre-prepared agro-market ontology or not and if they are found after listing the property of the word in the ontology it will extract that word from the string and couple it with its property and stores this information in a database so it could be used as an input for the crowd data analyzer component. The identified language is used to

select the pre-prepared agro-market ontologies, the Amharic one or the English one, specified in Annex A.

Crowd Data Analyzer Component

This component is responsible for analyzing and aggregating the information that is extracted from the crowd messages by the information extractor component. The component uses the temporary database that is created by the information extractor component as an input. The component is implemented by the **CrowdAnlyzer()**, a java class that performs the activities of the knowledge classification and ranking modules of this component.

The **CrowdAnlyzer()** java class has two methods that implements the two modules of this component. The methods are **Classifier()** and **Ranker()**.

Classifier()method is used for querying the temporary database, which is populated by the information extractor component, in order to select a product that is sold at a particular market place at a particular date and repeat this process until all the products in the temporary database are selected. The reason why we want to select in this manner is due to the fact that we are assuming that a given product sold at a particular market place is going to have the same or fall in a small rage of price. The selected products will be then passed to the ranker method along with the retrieved price information.

The **Ranker()** method accepts the list of products along with their particular prices from the classifier() method and performs aggregation on the prices of a particular product and come out with a range of prices in which most of the particular prices fall into. The output of this module, which is an average range of price for a particular product along with the date, will be stored as the agro-market price information.

The aggregation is implemented in this method by using the expectation-maximization (EM) algorithm of the Dawid and Skene's work [62]. We used the Majority Decision (MD) approach and modified the algorithm. Listing, and used it to calculate the average price of a given product from the suggested prices of the crowed

5.3.2.4 Crowd Manager Component

The objective of this component is to manage the crowds who participate in providing agro-market information to the system by sending an SMS message. What we mean by managing is, by keeping a profile or record of every crowder which sends an SMS message we are going to

give them motivation so that they want to participate again. The motivation is going to be done by giving them an air time or what we call traditionally "*mobile card*".

As some users might send the system a false data in order to get the motivation, this component also performs cheating detection by giving a trust level for every crowder who sends an SMS message which will decrease and increase depending on the information they provided compared to the other crowder, who provided the agro-market information.

This component is implemented in the prototype by the java class **crowdMan()**. The crowdMan() class has two methods that handle the activities of the component. They are**CrowdLevelController()** and**CrowdMotive()**.

CrowdLevelController() is a method that accepts the list of the senders who sent correct price values from the crowd data analyzer class in the form of an Array and selects all the sender numbers from the SMS directory database. It then compares the two lists and sets a flag of number "1" to the senders who sent in correct information and a flag of number "0" to the senders with the wrong price information.

After setting the flag the method will select the list of senders that have participated in sending an SMS message previously along with their trust level from the Crowd Directory database. The method then searches the list of senders with the flag in the list of senders with levels.

If a flagged sender with flag "0" is found in the list from the crowd directory database, then its corresponding trust level value will decrease by 1. For example if its trust level was 3 its new trust value will be 2. If a flagged sender with flag "1" is found in the list from the crowd directory database, then its corresponding trust level value will increase by 1. If a flagged sender is not found in the list from the crowd directory database, then a new trust level will be assigned to the sender according to the flag value. If flag value is 0 then the trust level will be 0 and if the flag is 1 then the trust value assigned to the sender will be 1.

Finally the method will update the crowd directory database with the new trust level for the senders

CrowdMotive() method undertakes the motivational task of the crowd management component. The method first selects sender's phone number and his/her corresponding trust level from the crowd directory database. If the trust level is 5 it will send an SMS message to the Crowder phone number saying "*You have got a 10 Birr card for your participation. Thank*

you" as depicted in Figure 5.2 and sends an air time worth 10 Birr to the crowder's phone. If the trust level is different from 5 it will send an SMS message to the Crowder phone number saying "*Your message has been delivered. Thank you for participating*" as depicted in Figure 5.3

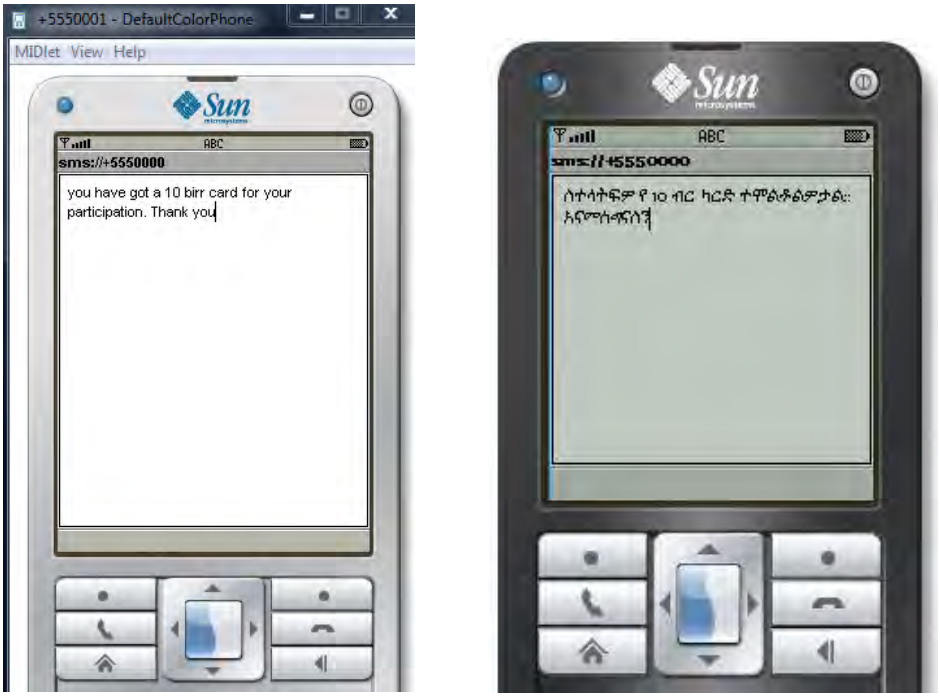

Figure 5.2: Screen Shot of an Emulator Receiving Air Time as Motivation in English and Amharic

Figure 5.3: Screen Shot of an Emulator Receiving Acknowledgment

Request Manager Component

The objective of this component is, once the agro-market information accepted from the crowd is processed and stored in the Agro Market Knowledge database, it provides this Agro Market information to a user that requests it through a request SMS message. The request message has to start with string “#req#” and then followed by the agricultural product name, so that the system will understand it as being a request SMS message and not a Crowd message.

The component is implemented in the prototype by a Java class **RequestMan()**. The java class has two methods named **Reqidentifier()** and **DataRetriver()**.

When a user sends a request SMS message the **SMSManagerServelet()** class accepts it and identifies it as being a request, because the message will have “**#req#**” string at the beginning, then calls the **Reqidentifier()** method with the message as an argument as shown in Figure 5.4.

The **Reqidentifier()** method accepts the message and stores the product name in a string and constructs an SQL query syntax to select the product from the Agro Market Knowledge database. Then the query will be executed and it will return the product name, price and the place it is sold. The retrieve data will be converted to a string and will be sent to the **DataRetriver()** method.

The **DataRetriver()** method accepts the string from the **Reqidentifier()**method and changes it to an SMS format by appending the senders number. Then this method will return the SMS message to the SMSManagerServelet() class which finally delivers it to the request senders mobile as shown in Figure 5.5.

Figure 5.4: Screen Shot of an Emulator Sending a Request Message

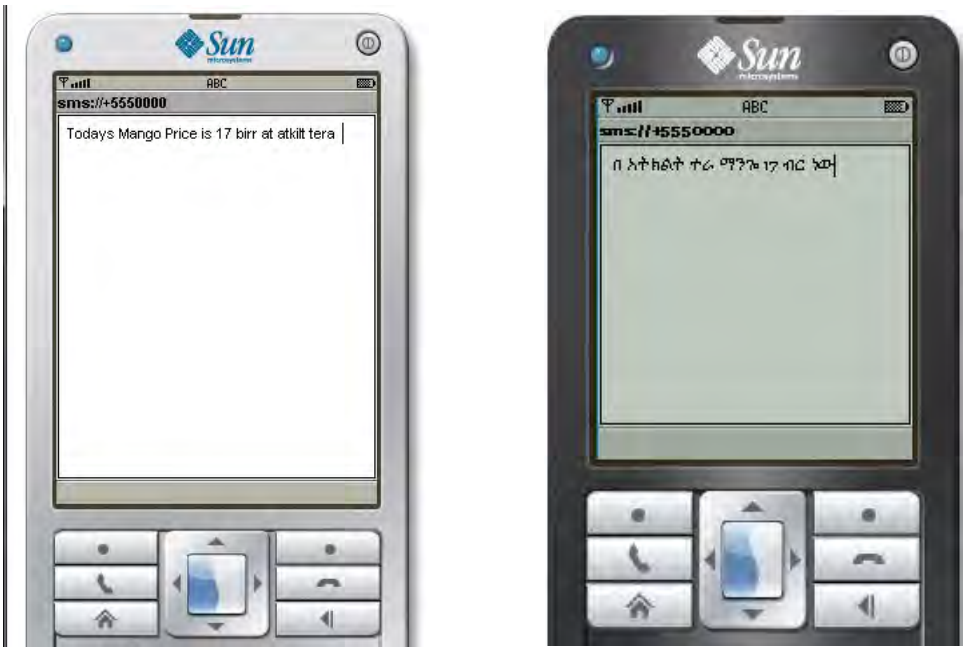

Figure 5.5: Screen Shot of an Emulator Receiving an Answer to Request

Chapter Six: Evaluation

6.1 Overview

In this Chapter we are going to evaluate the performance of the prototype we have developed for the proposed mobile based crowdsourcing for agricultural market information system. The evaluation is done to test if the prototype has implemented the proposed system requirements that are described in Chapter Four.

In order to evaluate the developed prototype, we have used 300 sample SMS messages, which we prepared, that have agro market information and presented in Annex C partially. Out of the 300 sample questions 150 are in Amharic and the rest are in English. The evaluation is done on the five basic components of the proposed system by using the sample SMS messages and calculating the performance of the components using precision and recall values as presented in the following sections.

6.2 SMS Manager Component Evaluation

The task of this component was to accept incoming SMS messages from the crowd mobile phones by the server and store them in the SMS directory database, accept a request SMS message from the user and pass it to the request manager component and finally accept the processed request result from the request manager component and send it to the requesters phone in the form of an SMS message.

The component was implemented with a Java servlet class that resides in the server and communicates with the crowder's phone using the HTTP communication. The test for this component was done on a personal computer which has 2.3 GHz processing speed acting as a server.

By using the sample 300 SMS messages we have evaluated this component. All the incoming messages were accepted by the server and stored in the database or forwarded to the request manager component and the returning SMS messages were sent to the user's phone.

The connection with the emulator and server was successfully implemented by the prototype.

6.3 Information Extraction component Evaluation

In order to evaluate this component we have used the sample 300 SMS messages prepared by the researcher and calculated the precession and recall of the system language identification. We didn't use actual users to test the prototype due to the fact that the prototype was developed in a simulation environment. The result of the evaluation is shown in Table 6.1.

Table 6.1: Precession and Recall for Language Identification

Language identified	**precession**	**Recall**
Amharic	86.5%	84.3%
English	92.4%	91.7%

As Table 6.1 shows, the precision and recall of identifying the language of the message as Amharic is 86.5% and 84.3% respectively. While identifying the language of the message as English shows 92.4% precision and 91.7% recall. The reason why the Amharic has low precision and recall is due to the fact that numbers and punctuations are taken as an English character instead of being an Amharic one.

To evaluate the implementation of the information extraction module of this component in the prototype, we have measured the precession and recall of extracting the meaning of the words in the message from the ontology and it is shown in Table 6.2.

Table 6.2: Precession and Recall for Information Extraction

Class identified	**Precession**	**Recall**
Product	90.67%	87.4%
Place	87.3%	85.91%
Price	91.5%	89.26%

As Table 6.2 shows, the precision and recall of extracting the meaning of the words of the message using ontology is more than 80%. The errors that occurred in the testing stage are due to the fact that our ontology is a static one and words that are new or that have spelling errors

couldn't have class inferred for them. As Date value is taken from the time the SMS was received by the Server, we didn't include it in this evaluation.

6.4 Data Analysis Component Evaluation

We have evaluated this component by extracting the information from the sample 300 SMS messages and group the same product types together for the Expectation Maximization algorithm to process then we measured the precession and recall of the aggregation process varying the number of product price processed at a time and the result is shown in Table 6.3.

Table 6.3: Precession and Recall for the Number of Records to Aggregate

Number of records processed at one time	**Precession**	**Recall**
10 records	63%	58%
20 records	67%	64.5%
30 records	72%	70%
>30 records	68%	67.5%

As shown in Table 6.3, the precession and recall for processing 30 records is higher than the others. This is due to the fact that the algorithm performs well and calculates the most precise price average range when the number of record is not too big or not too small.

6.5 Request Management Component Evaluation

We evaluated the performance of this component that is implemented in the prototype by taking 50 sample request messages that are included in the 300 sample testing SMS messages. And we found that the system gives a response with accuracy of 97%.

Chapter Seven: Conclusion and Future work

7.1 Conclusion

We have proposed and implemented a mobile based crowdsourcing system for agro market information in this thesis work. The aim of our proposed system is to process and provide aggregated and updated agricultural market price information which is provided by the crowd using SMS messaging that is available in Java based mobile phone devices.

Our proposed system has five basic components which are implemented in a server. The first component is the SMS manager component, which is responsible for accepting and sending SMS messages to and from the server and the users or crowd mobile phone. The second one is the information extraction component which uses ontology based information extraction approach to extract basic agro market information which are; product name, place the product is being sold, the date on which the product was sold and the price of the product from the crowd.

The third component is the data analysis component which aggregates the price of a given product from the table created by the information extraction component and generates the agro market knowledge database which is the required outcome of the proposed system. The fourth component is the crowd management which is responsible for managing the crowd database and giving reward. The last component is the request management component which accepts a request from a user who wants to know the price of a product and the component processes the request and returns the result in the form of an SMS message.

We have implemented the proposed system by developing a prototype. After evaluating the performance of the prototype we have got 90% accuracy in the information extraction and 75% accuracy in generalizing and aggregating the extracted information. So we can say that our work can be implemented to give a help in the agricultural economic development.

7.1 Contribution of the Work

Besides designing a new mobile based crowd sourcing agro market information system architecture, we have the following contribution:

- The study has created a system that can be implemented using the existing SMS messaging service.
- The study has successfully combined mobile based crowdsourcing with information extraction.
- The study has implemented ontology based information extraction on SMS messages.
- The study has showed that crowdsourcing can be used to exchange market price information in a collaborative manner.

7.2 Future Work

As a future work, we recommend the following works to be done in the future as additional investigation and implementation to improve the performance of the proposed system.

- The implementation of dynamic ontology, so that new products and places can be added to the ontology and increase the information extraction process
- Using natural language processing along with ontology approach in order to increase the performance of the information extraction process.
- Using multilingual ontology to process SMS messages written in different language.
- Implementing the system to be used by all type of mobile phones, not only java based mobiles but also android based and IOS based mobile phones

References

[1] Amanuel Z, Komminist W., Sebsibe H,"Towards Designing an Architecture for Delivering Distributed Agricultural Information Services for Developing Countries: Technical report, Fondazion Bruno Kessler, Trento, Italy,2011.

[2] Howe J., "Crowdsourcing: Why the Power of the Crowd is Driving the Future of Business". Crown Business, 1st Edition, August, 2008.

[3] Daren C. Brabham, "Crowdsourcing as a Model for Problem Solving", The International Journal of Research into New Media Technologies, Vol 14(1): 75–90,2008.

[4] EU Commission:"Studies on Translation and Multilingualism, Crowdsourcing Translation", 2011.

[5] Introduction to iStockphoto:http://www.istockphoto.com/introduction.php, last accessed on December, 2013.

[6] About InnoCentive, http://www.innocentive.com/about/index.html, last accessed on November,2012.

[7] Joseph C.,"Crowdsourcing a Personalized Learning Environment for Mathematics Knowledge", Media Institute, the Open University, Milton Keynes, UK.

[8] Wikipedia, https://en.wikipedia.org, last accessed on June, 2014.

[9] Ramirez, J. Ushahidi technology saves lives in Haiti and Chile. Retrieved fromNewsweek:http://blog.newsweek.com/blogs/techtonicshifts/archive/2010/03/03/ushahidi-technology-saves-lives-in-haiti-and-chile.aspx last accessed on November,2012.

[10] About Threadless, http://threadless.com/about, last accessed on November, 2012

[11] Eagle, N., "txteagle: Mobile Crowdsourcing", In Internationalization, Design and Global Development, Volume 5623, pp 447-456, 2009.

[12] Slavova, M., "ICT Update - A Current Awareness Bulletin for ACP Agriculture". Retrieved from ICT Update: http://ictupdate.cta.int/en/Feature-Articles/Encouraging-foreign-exchange,last accessed on July,2013.

[13] Surowiecki J., "The Wisdom of Crowds: Why the Many are Smarter than the Few and How Collective Wisdom Shapes Business, Economies, Societies, and Nations", New York, Doubleday, 2004.

[14] Man-Ching Y., Irwin K., and Kwong-Sak L., “A Survey of Crowdsourcing Systems”, IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing, 2011.

[15] Florian A., Alireza S., Albrecht S., Urs K., and Zahid N., “ Location-based Crowdsourcing: Extending Crowdsourcing to the Real World”,*NordiCHI 2010*, Reykjavik, Iceland, 2010.

[16] M.-C. Yuen, I. King, and K.-S. Leung, “Task Matching in Crowdsourcing”, In *CPSCom ’11: Proceedings of the 4th IEEE International Conferenceson Cyber, Physical and Social Computing*, IEEE Computer Society, 2011.

[17] L. von Ahn, S. Ginosar, M. Kedia, R. Liu, and M. Blum, “Improving Accessibility of the Web with a Computer Game”, In *ACM SIGCHI Conference on Human Factors in Computing Systems*, 2006.

[18] V. Hester, A. Shaw, and L. Biewald, “Scalable crisis relief: Crowdsourced SMS Translation and Categorization with Mission 4636”, ACM DEV’10, London, 2010.

[19] Amazon Mechanical Turk, https://www.mturk.com/, last accessed on May, 2013.

[20] Yahoo’s Flickr, http://www.flickr.com/, last accessed on February, 2013.

[21] A. Kittur, B. Smus, and R. E. Kraut, “CrowdForge: Crowdsourcing Complex Work”, 24th annual ACM symposium on User interface software and technology, ACM, pp. 43-52, 2011.

[22] AboutInnoCentive, http://www.innocentive.com/about/index.html last accessed on February, 2013.

[23] CambrianHouse, https:// www.cambrianhouse.com/, last accessed on March, 2014.

[24] LauriPitkänen and Joni Salminen, “Crowdsourcing Research to Mobile Consumers? Emerging Themes on Videographic Data Collection”, International Conference on Management, Applied and Social Sciences, ICMASS'2012, Dubai, 2012.

[25] Konomi, S., Thepvilojana N., Suzuki, R., Pirttikangas S., Sezaki K., and Tobe Y., “Askus Amplifying Mobile Actions”, Procedural Pervasive, 2009.

[26] Fashism, https:// www.fashism.com, last accessed on February, 2013.

[27] Google Map, https://www.google.com/mobile/products/maps.html, last accessed on February, 2014.

[28] Okolloh O., “Ushahidi, or 'testimony': Web 2.0 Tools for Crowdsourcing Crisis Information”, Participatory Learning and Action, No. 59, 2009.

[29] B. Mellebeek, F. Benavent, J. Grivolla, J. Codina, M. R. Costa-juss`a, and R. Banchs, "Opinion Mining of Spanish Customer Comments with Nonexpert Annotations on Mechanical Turk", In *Proceedings of the NAACLHLT 2010 Workshop on Creating Speech and Language Data with Amazon'sMechanical Turk*, CSLDAMT '10, pp. 114–121, USA, 2010.

[30] L. von Ahn, M. Kedia, and M. Blum, "Verbosity: A Game for Collecting Common-Sense Facts", *ACM SIGCHI Conference on Human Factors in Computing Systems*, 2006.

[31] H. Lieberman, D. Smith, and A. Teeters, "Common Consensus: A Web Based Game for Collecting Commonsense Goals", *ACM Workshop on Common Sense for Intelligent Interfaces*, 2007.

[32] O. Alonso, D. E. Rose, and B. Stewart, "Crowdsourcing for relevance evaluation", *SIGIR Forum,* No 42, 2008.

[33] C. Akkaya, A. Conrad, J. Wiebe, and R. Mihalcea, "Amazon Mechanical Turk for Subjectivity Word Sense Disambiguation", In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, CSLDAMT '10*, USA, 2010.

[34] Q. Gao and S. Vogel, "Consensus versus Expertise: A Case Study of Word Alignment with Mechanical Turk", In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, CSLDAMT '10*, USA, 2010.

[35] M. Jha, J. Andreas, K. Thadani, S. Rosenthal, and K. McKeown, "Corpus Creation for New Genres: A Crowdsourced Approach to attachment", In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, CSLDAMT '10,* USA, 2010.

[36] G. Parent and M. Eskenazi, " Clustering Dictionary Definitions using Amazon Mechanical Turk", In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, CSLDAMT '10,* USA, 2010.

[37] OpenStreetmaps, https:// www.openstreetmap.org,lastaccessed on December, 2014.

[38] iStockphoto, https://www.istockphoto.com/,last accessed on February, 2013.

[39] L. von Ahn, "Games with a Purpose", *IEEE Computer*, No 39, pp 92–94, June 2006.

[40] Foldit, https://www.fold.it, last accessed on February, 2013.

[41] L. von Ahn and L. Dabbish,"Labeling Images with a Computer Game", *ACM SIGCHI Conference on Human Factors in Computing Systems*, 2004.

[42] E. Law and L. von Ahn, “Input-agreement: A new Mechanism for Data Collection Using Human Computation Games, *ACM CHI*, 2009.

[43] Threadless, http://www.Threadless.com,last accessed on February, 2013.

[44] Mason W. and Watts D. J, “Financial Incentives and the Performance of Crowds”, Proc. HCOMP '09, 2009.

[45] J. S. Downs, M. B. Holbrook, S. Sheng, and L. F. Cranor, “Are Your Participants Gaming the System?” In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI ’10, USA, 2010.

[46] B. A. Huberman, D. M. Romero, and F. Wu, “Crowdsourcing, Attention and Productivity”, *Information Science*, No 35: pp 758–765, 2009.

[47] C. G. Harris, “You’re Hired! An Examination of Crowdsourcing Incentive Models in Human Resource Tasks”, In *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM 2011)*, pp 15–18, 2011.

[48] M. S. Silberman, L. Irani, and J. Ros, “Ethics and Tactics of Professional Crowdwork”, *XRDS*, no 17, pp 39–43, 2010.

[49] K. K. Nam, M. S. Ackerman, and L. A. Adamic, “Questions in knowledge ? : A Study of Naver’s Question Answering Community”, In *Proceedings of the 27th international conference on Human Factors inComputing Systems*, CHI ’09, pp 779–788, 2009.

[50] DiPalantino D. and Vojnovic M., “Crowdsourcing and Allpay Auctions”, 10th ACM conference on Electronic commerce, ACM, pp. 119-128, 2009.

[51] P. G. Ipeirotis. “Analyzing the Amazon Mechanical Turk Marketplace, *XRDS*, No 17, pp 16–21, 2010.

[52] Grady and M. Lease, “Crowdsourcing Document Relevance Assessment with Mechanical Turk”, In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, CSLDAMT ’10, pp 172–179, 2010.

[53] J. Wang, S. Faridani, and P. G. Ipeirotis, “Estimating the Completion Time of Crowdsourced Tasks Using Survival Analysis Models”, In *Proceedingsof the Workshop on Crowdsourcing for Search and Data Mining (CSDM2011)*, pp. 31–34, 2011.

[54] P. G. Ipeirotis, F. Provost, and J. Wang., “Quality Management on Amazon Mechanical Turk”, In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP ’10, pp 64–67, 2010.

[55] B. Carterette and I. Soboroff, “The Effect of Assessor Error on IR System Evaluation”, In *Proceeding of the 33rd International ACM SIGIRConference on Research and Development in Information Retrieval*, SIGIR ’10, pp. 539–546, 2010.

[56] S. Jain and D. C. Parkes, “The Role of Game Theory in Human Computation Systems”, In *Proceedings of the ACM SIGKDD Workshop on HumanComputation*, HCOMP ’09, pp. 58–61, 2009.

[57] D. Feng, S. Besana, K. Boydston, and G. Christian, “Towards High Quality Data Extraction via Crowdsourcing”, In *Proceedings of the CrowdConf ‘10*, 2010.

[58] A. Kittur and R. E. Kraut, “Harnessing the Wisdom of Crowds in Wikipedia: Quality Through Coordination”, In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work*, CSCW ’08, pp. 37–46, 2008.

[59] C. Eickhoff and A. P. de Vries, “How Crowdsourcable is Your Task?”, In *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM 2011)*, pp. 11–14, 2011.

[60] M. Hirth, T. Hoßfeld, and P. Tran-Gia, “Cheat-detection Mechanisms for Crowdsourcing”, University of W. urzburg, 2010.

[61] V. Almendra and D. Schwabe, “Fraud Detection by Human Agents: A Pilot Study”, In *Proceedings of the 10th International Conference on E-Commerce and Web Technologies*, EC-Web 2009, pp. 300–311, 2009.

[62] Dawid, A.P. and Skene, A.M., “Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm”, Applied Statistics, Vol. 28, No. 1, pp. 20-28. 1979.

[63] A. Gupta, W. Thies, E. Cutrell, and R. Balakrishnan, “mClerk: Enabling Mobile Crowdsourcing in Developing Regions”, ACM CHI, pp. 1843-1852, 2012.

[64] P. Narula, P. Gutheim, D. Rolnitzky, A. Kulkarni, and B. Hartmann, “MobileWorks: A Mobile Crowdsourcing Platform for Worker at the Bottom of the Pyramid”, University of California, Berkeley, 2011.

[65] R. Cooper, S. Ali and C. Bi, “Extracting Information from Short Messages”, NLDB’05 proceding of the 10th International Confrence on Natural Language Processing and Infromation Systems, pp. 388- 391, 2005.

[66] D. Muhammad and D. Hendratmo, “Information Extraction from Short Text Message in Bahasa Indonesia for Electronics”, JurnalSarjanaInstitutTeknologi Bandung bidangTeknikElektrodanInformatik, Vol. 1, No. 1, April 2012.

Annexes

A: The ontology developed to be used in the information extraction component

<rdf:RDF

xmlns:AgreProduct="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#"

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

xmlns:owl="http://www.w3.org/2002/07/owl#"

xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"

xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#mango">

<rdf:type
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#atakelt_ena_frafre"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Thing"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#merkato">

<rdf:type
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Everyday_Markets"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Thing"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Supermarkets"
>

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Market"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Shops">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Market"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#sende">

<rdf:type
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Tra_tere"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Thing"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#tefe">

<rdf:type
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Tra_tere"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Thing"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Everyday_Markets">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Market"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Dereja">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#ByQuality"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#orange">

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Thing"/>

<rdf:type
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#atakelt_ena_frafre"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Region">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Location"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Price">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#AgroMarket
"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl">

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Ontology"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Product">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#AgroMarket
"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#YegeterKebele"
>

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Location"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Wofchobets">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Market"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#ye_kebat_ehloch">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Product"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Wereda">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Location"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#kuntal">

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Thing"/>

<rdf:type
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#ByQuantity"
/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#atakelt_ena_frafre">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Product"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Measurement"
>

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#AgroMarket
"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Market">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Place"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#gulet">

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Thing"/>

<rdf:type
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Day_Markets"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#bekolo">

<rdf:type
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Tra_tere"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Thing"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Day_Markets">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Market"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#birr">

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Thing"/>

<rdf:type
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Price"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Tra_tere">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Product"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#ByQuantity">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Measurement"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#City">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Location"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#AgroMarket">

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#ByQuality">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Measurement"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Location">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Place"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Place">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#AgroMarket
"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#banana">

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Thing"/>

<rdf:type
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#atakelt_ena
_frafre"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#ye_seble_mirto
ch">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Product"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

<rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#kefleKetema">

<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/1/AgreProduct.owl#Location"/>

<rdf:typerdf:resource="http://www.w3.org/2002/07/owl#Class"/>

</rdf:Description>

</rdf:RDF>

B: Proto type Implementation Sample java Code

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throwsServletException, IOException {

        int length = request.getContentLength();
        byte[] input = new byte[length];
        ServletInputStream sin = request.getInputStream();
        int c, count = 0 ;
int i=0;
  String x[] =  new String[20];
        while ((c = sin.read(input, count, input.length-count)) != -1) {
        count +=c;
                }
        sin.close();

                String recievedString = new String(input);

System.out.println(recievedString);
int start = 6;
int end = 14;
charfrm[] = new char[end - start];
recievedString.getChars(start, end, frm, 0);

String from= new String(frm);
System.out.println(from);
start = 14;
end =24;
char d[] = new char[end - start];
recievedString.getChars(start, end, d, 0);
String date= new String(d);
System.out.println(date);

start = 25;
```

```
end =33;
char t[] = new char[end - start];
recievedString.getChars(start, end, t, 0);
String time= new String(t);
System.out.println(time);
start = 33;
end =recievedString.length();
char mg[] = new char[end - start];
recievedString.getChars(start, end, mg, 0);
String msg= new String(mg);
System.out.println(msg);
      Connection con = null;
Statement stmt = null;
ResultSetrs = null;
response.setContentType("text/html");
PrintWriter out = response.getWriter();
try {
// Load (and therefore register) the Oracle Driver
Class.forName("com.mysql.jdbc.Driver");
// Get a Connection to the database
con = DriverManager.getConnection(
"jdbc:mysql://localhost:3306/crowdmessage", "root", "root");
// Create a Statement object
stmt = con.createStatement();
String sql = "INSERT INTO MSG (Date,Time,Telephone,MessageRecived) " + "VALUES (' "+date+"
',' " +time +" ',' "+from+" ',' " +msg +" '  )";
stmt.executeUpdate(sql);
System.out.println("succesfuly updated database");
ResultSetsrs = stmt.executeQuery("SELECT MessageRecived  FROM MSG");

while(srs.next())
 {
  //for (int i=0; i<=20 ; i ++)
 // {
```

```
x[i]= srs.getString("MessageRecived");
System.out.println( x[i]);
i++;
   //}
 }
}
catch(ClassNotFoundException e) {
out.println("Couldn't load database driver: " + e.getMessage());
}
catch(SQLException e) {
out.println("SQLException caught: " + e.getMessage());
}
finally {
// Always close the database connection.
try {
if (con != null) con.close();
}
catch (SQLException ignored) { }
}
```

C: Sample SMS Messages used in Evaluation of the Prototype

1. Today Mango price is 30 birr in kilo around piassa
2. Teff price on merkato kuntal 1800
3. Coffee price for Yirgachefe in kilo is 65 birr
4. Today tomato price in atkilttera in kilo is 13 birr
5. Today onion price in Atkilttera in kilo is 10 birr
6. Banana price at Mexico in kilo 14 birr
7. Wheat price in Quntal today in Gojjamberenda is 560 birr
8. ዛሬ በመርካቶ ጤፍ በኩንታል 1800 ብር እየተሸጠ ነው
9. ዛሬ እህል በረንዳ 100 ኪሎ በቆሎ 870 ብር እየተሸጠ ነው
10. ሽንኩርት ዛሬ በአትክልት ተራ በኪሎ 13 ብር እየተሸጠ ነው
11. Orange price in etfruit in kilo is 35 birr
12. በስቢታ ዛሬ ኩንታል ገብስ በ580 ብር እየተሸጠ ነው
13. Banana 13 birr at etfruit
14. ገብስ 580 ብር እህል በረንዳ
15. The price of coffee at ECX is 65 birrs for a kilo.