DISSERTATIONS IN
FORESTRY AND
NATURAL SCIENCES

Andrei-Cătălin Tabarcea

Location-Based Web Search and Mobile Applications

Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences

UNIVERSITY OF
EASTERN FINLAND

ANDREI-CĂTĂLIN TABARCEA

Location-Based Web Search and Mobile Applications

Publications of the University of Eastern Finland

Dissertations in Forestry and Natural Sciences

Number 185

Academic Dissertation

To be presented by permission of the Faculty of Science and Forestry for public examination in the Auditorium M101 in Metria Building at the University of Eastern Finland, Joensuu, on September 30, 2015, at 12 o'clock noon.

School of Computing

Grano Oy

Jyväskylä, 2015

Editors: Dir. Pertti Pasanen,

Prof. Pekka Kilpeläinen, Prof. Kai Peiponen, Prof. Matti Vornanen

Distribution:

Eastern Finland University Library / Sales of publications

P.O.Box 107, FI-80101 Joensuu, Finland

tel. +358-50-3058396

http://www.uef.fi/kirjasto

ISBN: 978-952-61-1868-0 (printed)

ISBN: 978-952-61-1869-7 (pdf)

ISSNL: 1798-5668

ISSN: 1798-5668

ISSN: 1798-5676 (pdf)

Author's address: University of Eastern Finland
School of Computing
P.O.Box 111
80101 Joensuu
FINLAND
email: tabarcea@cs.uef.fi

Supervisors: Professor Pasi Fränti, Ph.D.
University of Eastern Finland
School of Computing
P.O.Box 111
80101 Joensuu
FINLAND
email: franti@cs.uef.fi

Reviewers: Helena Ahonen-Myka, Ph.D.
Tibidiscis Oy
Finnoonniitynkuja 4
02270 ESPOO
FINLAND
email: hahonen@cs.helsinki.fi

Dirk Ahlers, Ph.D
Norwegian University of Science and Technology
Department of Computer and Information Science
Sem Sælands vei 9
NO-7491 Trondheim
NORWAY
email: dirk.ahlers@idi.ntnu.no

Opponent: Professor Tapio Salakoski, Ph.D.
University of Turku
Department of Information Technology
ICT-talo, 6th floor, Joukahaisenkatu 3-5 B
20520 TURKU
FINLAND
email: tapio.salakoski@utu.fi

ABSTRACT:

Location is an important factor in personalizing applications in various fields such as web search, data mining, contextual recommendations or mobile gaming. Nowadays, due to the rapid development and wide availability of positioning techniques and internet connectivity, location is easily available and significantly improves the applications that utilize the user's context.

This thesis presents contributions in the field of location-based applications. It proposes applications and advances in location-based web search and data mining, postal address detection and location-based gaming. We verified our methods and algorithms using data collected by our users within the MOPSI project.

The first part of the thesis describes an application that identifies location-based data in web pages. Location data are widely available on the web, but rarely in a standardized format. Most of the time they are available as postal addresses, especially on the web pages that describe commercial services or points of interest. We are detecting addresses by using a gazetteer-based method. Our gazetteers use freely available data sources such as OpenStreetMap. Furthermore, we extract a title and a representative image for each detected location. Our goal was to provide information that is close to the user's location, related to the keywords provided by the user and extracted from the content of websites.

The second part of the thesis describes a location-aware mobile game that promotes physical exercise by applying concepts from the classical game of orienteering and uses a geo-tagged photo collection created by users.

The results of the work documented in this thesis were integrated into services that are available to users through mobile phone application or web pages.

Universal Decimal Classification: 004.738.5, 004.774, 004.775, 004.78, 004.9

Library of Congress Subject Headings: Mobile computing; Mobile apps; Location-based services; Wireless localization; Geographical positions; Global Positioning System; Data mining; Web sites; Internet; Street addresses; Mobile games; Orienteering; Cell phones; Smartphones

Yleinen suomalainen asiasanasto: mobiilisovellukset; mobiilipalvelut; mobiililaitteet; paikannus; tiedonlouhinta; WWW-sivut; Internet; osoitteet; mobiilipelit; suunnistus; matkapuhelimet; älypuhelimet

INSPEC Thesaurus: location-based application; web data mining; mobile game; GPS; orienteering; postal address detection

Acknowledgments

First of all, I would like to extend my sincerest gratitude to Professor Pasi Fränti, head of the Speech and Image Processing Unit from the School of Computing at the University of Eastern Finland, for his support and help with research throughout the years. None of my accomplishments and experience gained within these years would have been possible without his guidance. I would also like to thank all the members of the Speech and Image Processing Unit, the administrative staff or School of Computing and all the people that worked within the MOPSI project for creating a great working atmosphere and for providing help and support with my work whenever I needed it. I have learned a lot from all of you.

I am also grateful to Professor Vasile Manta and the Technical University of Iaşi for supporting me and for providing the opportunity to study and research abroad through the Erasmus program and through the joint doctoral agreement.

Last but not least, I would like to thank my family, friends and Cristina for their moral support and care.

This research has been supported by the East Finland Graduate School in Computer Science and Engineering (ECSE), the Technical University "Gheorghe Asachi" of Iaşi, the University of Eastern Finland, the Nokia foundation, and the MOPSI and MOPIS projects. All their support is gratefully acknowledged.

Joensuu September 3, 2015 *Andrei Tabarcea*

LIST OF ABBREVIATIONS

DOM	Document-Object Model
GIS	Geographic Information Systems
GPS	Global Positioning System
HTML	HyperText Markup Language
LBS	Location-Based System
MOPSI	Mobiilit Paikkatieto-Sovellukset ja Internet (Mobile location-based applications and Internet)
URL	Uniform Resource Locator

LIST OF PUBLICATIONS

This thesis presents a current review of the author's work in the field of location-based applications, and the following selection of the author's publications:

[P1] P. Fränti, J. Chen, A. Tabarcea, "Four aspects of relevance in location-based media: content, time, location and network", Int. Conf. on Web Information Systems and Technologies (WEBIST'11), Noordwijkerhout, Netherlands, 413–417, May 2011.

[P2] P. Fränti, A. Tabarcea, J. Kuittinen, V. Hautamäki, "Location-based search engine for multimedia phones", IEEE Int. Conf. on Multimedia and Expo (ICME'10), Singapore, 558–563, July 2010**.**

[P3] A. Tabarcea, V. Hautamäki, P. Fränti, "Ad-hoc georeferencing of web-pages using street-name prefix trees", Int. Conf. on Web Information Systems and Technologies (WEBIST'10), Valencia, Spain, vol.1, 237–244, April 2010.

[P4] A. Tabarcea, N. Gali, P. Fränti, "Location-aware information extraction from the web" (manuscript), 2015.

[P5] N. Gali, A. Tabarcea, P. Fränti, "Extracting representative image from web page". Int. Conf. on Web Information Systems and Technologies (WEBIST'15), Lisbon, Portugal, May 2015.

[P6] A. Tabarcea, K. Waga, Z. Wan and P. Fränti, "O-Mopsi: Mobile Orienteering Game Using Geotagged Photos", Int. Conf. on Web Information Systems and Technologies (WEBIST'13), Aachen, Germany, 8–10 May 2013.

The original publications are included at the end of this thesis by permission of their copyright holders. Throughout the overview, these papers will be referred to as [**P1] –[P6]**.

AUTHOR'S CONTRIBUTION

The contributions of the authors of these papers to this dissertation can be summarized as follows: In [**P1**] the authors define four aspects of relevance in sharing location-based media: location, time, content and social network and study how they appear in media sharing platforms. Prof. Pasi Fränti wrote the paper, Jinhua Chen developed the web interfaces and the author implemented the mobile software, contributed to text writing and performed all experiments for this paper.

[**P2]** describes a location-aware search engine for web and mobile environment. It sketches the overall scheme of the MOPSI search engine prototype, defines all the needed core elements and tests a prototype for Finland. The idea was proposed by Prof. Pasi Fränti. The first draft of the search engine was developed jointly by the author and Juha Kuittinen, but the author was responsible for the version used in this paper, performed all mobile-side programming and experiments and also is the main contributor in writing Sections 3 and 4.

[**P3]** describes and tests the address detection algorithm in [**P2]**, which is based on individually detecting address elements and aggregating them as address candidates that are validated using gazetteers. **[P4]** improves [**P2]** and [**P3]** by replacing plain text extraction with processing of the DOM representation and by improving the methods for extracting the title and representative image for each search result.

For [**P3]** and **[P4]**, the author was the main contributor for the development of ideas and technical solutions, and was the sole person to implement all the related mobile applications. He performed all experiments and was responsible for writing the paper. Other authors mostly provided supported mostly by means of advice and text revisions.

[**P5]** studies how to select a representative image to represent an entire web page. The authors propose a rule-based method to categorize images based on their purpose in the web page. This solution is needed as part of the summarization of the web page

found by MOPSI search. The paper is a result of team work where Najlah Gali and the author jointly contributed to the idea development. The author contributed to the implementation of the proposed method, the experimentation, and paper writing.

Finally, **[P6]** described a location-based mobile orienteering game that aims to promote physical exercise and learn new technologies. The game is based on user-generated data from our MOPSI system and was presented during a yearly international festival in which middle- and high-school students learn about science, technology and the environment. Prof. Pasi Fränti contributed with the idea, Karol Waga and Zhentian Wan made the web implementation, and the author created the mobile solutions. The author also wrote the paper and did all experimentations; all authors contributed to organizing the O-Mopsi workshop in SciFest, where the test material was collected.

Contents

1 Introduction

Exploiting the users' geographical location has become more and more popular during recent years, mainly because of the increasing availability of GPS enabled mobile devices such as smartphones or personal navigators and the constant decrease in the prices of such devices. Additionally, extra positioning methods such as cellular network positioning and Wi-Fi positioning facilitate the access to the users' location. Therefore, during the last years there has been increasing interest in the research of location-based services, both in academic and commercial projects.

A location-based service is an application which integrates the user's geographical location with the general notion of service with the purpose to provide information about a certain place or geographical location [ScVo04]. Usually, location-based services are accessible through mobile devices connected to a mobile network and they use the location information provided by the mobile device. There are many categories of location-based services, such as: navigation, search and providing information, monitoring, advertising, management, games, socializing etc. A location-based application is an application that uses such services.

Location-based services are part of the larger field of context-aware services, which are services that adapt their way of functioning according to one or more parameters which reflect the context of targets or users [Küpp05].

Location-based data are very common on web-pages, especially when their content describes commercial services, landmarks or public institutions. However, the location data are rarely embedded as geographical coordinates that can be retrieved automatically, but are more commonly presented in a

human-readable way that can be retrieved using location-based web search.

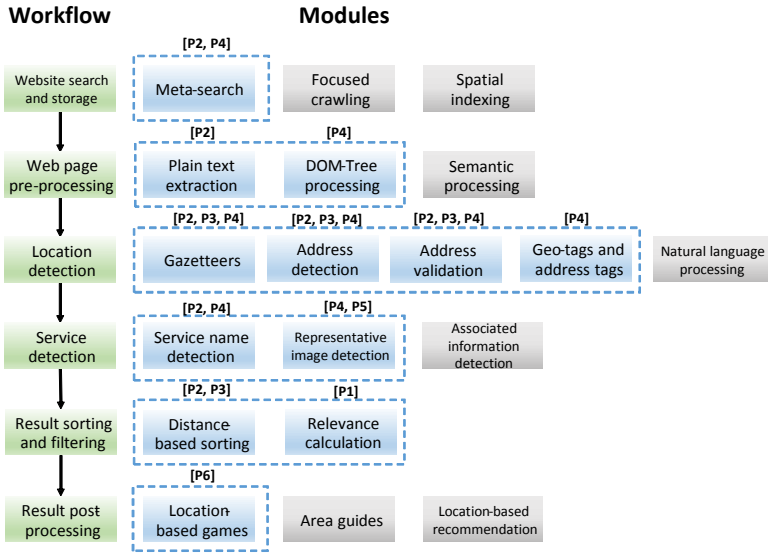

Figure 1-1 Typical workflow and modules for a location-based search solution

As shown in ***Figure 1-1*** (left), a typical workflow of a location-based search solution requires the following steps: website search and storage, web page pre-processing, location detection, service detection, result sorting and filtering, and, optionally, result post-processing. These steps are the research topics covered in this thesis. ***Figure 1-1*** (right) shows possible approaches for each of the proposed steps, out of which we highlighted the modules that are covered in this thesis and in the papers that support it.

As positioning is ubiquitous in our electronic devices and more and more location data are produced every day, there is an increasing need to exploit the location data on the web and to provide results that are relevant in a specific location and are presented to the user in a clear and informative way. In this dissertation, we present a solution that is based on location data and has two types of applications: location-based web search and mobile applications. Our location-based web search solution proposes a method to identify location data from websites by detecting postal addresses. Our mobile applications are: a

location-based search solution for multimedia phones [P2], an application for collecting location-based data [P1] and a mobile game based on the concept of orienteering [P6]. Our mobile game, O-Mopsi, shows an example of how to use location data after they have been identified using location-based search or after they have been collected by mobile applications. Our solution and applications have been integrated into a location-based platform called MOPSI.

1.1 MOPSI PROJECT

The work in this thesis has been carried out within the MOPSI project[1], which is a research project for location-based services that is developed by the Speech and Image Processing Group from the School of Computing at the University of Eastern Finland. MOPSI offers multiple uses of location-aware applications, being a test-bed for various research topics that involve location-aware data. It contains tools for collecting, processing and displaying location-based data, such as photos or trajectories, along with social media integration.

The main topics addressed in MOPSI are: collecting location-based data, mining location data from web pages, processing, storing and compressing GPS trajectories, detecting transportation mode from GPS trajectories, recommending points of interest, using location information in social networks, and detecting users' actions by using their location and building location-based games with the help of user-generated collections.

MOPSI provides tools to collect GPS trajectories and our collection includes more than seven million GPS points, which are assigned to more than 8.000 trajectories. We designed a system for fast retrieval and displaying of the data [WTMF13] that is based on GPS trajectory polygonal approximation [ChXF12]. GPS trajectories are also compressed for optimizing storage. Furthermore, transport mode information can be

[1] http://cs.uef.fi/mopsi

retrieved from automatically analyzing GPS trajectories. We are using a second order Markov model to segment the trajectories and to detect stops, bicycle, running, or car transportation modes [WTCF12]. Furthermore, we have developed a system that calculates the similarity of GPS trajectories using a low complexity spatial measure [MTSF14].

The relevance of location-based media can be assessed by considering several aspects such as time, location, content or social network [P1], which are used to create a context for each user. Using our applications, users can collect geo-tagged photos; our collection includes more than 35.000 photos. A personalized recommendation system can recommend relevant data based on user location and on user context [WaTF12]. Such data can be geo-tagged photos, services confirmed by administrators or GPS trajectories.

Users can share their location in real-time by using mobile phone location-aware applications. This allows for the detection of various location-based actions such as meetings, visiting or passing-by points of interests [Mari13].

MOPSI also includes location-based games, such as O-Mopsi [P6], [Wan14], which describes how to create an orienteering game using the data from a user-generated photo collection and how to develop a web interface and a mobile application.

MOPSI provides tools for collecting location-based data with mobile devices. It is available on most of the major mobile operating systems (Android, iOS, Windows Phone, Symbian). On the server-side we process and display the data collected by users and also provide social features and integration with social media, with functionalities such as chatting, friend tracking and sharing data to Facebook.

1.2 FOUR ASPECTS OF RELEVANCE IN LOCATION-BASED MEDIA

Location-based services are becoming widely used due to the fast development of positioning systems in multimedia phones. Location provides additional information that can be expressed as a point of interest, route or geographic area. Location itself can be considered as information, but it is often attached to other data and shared via location-based services or photo sharing sites.

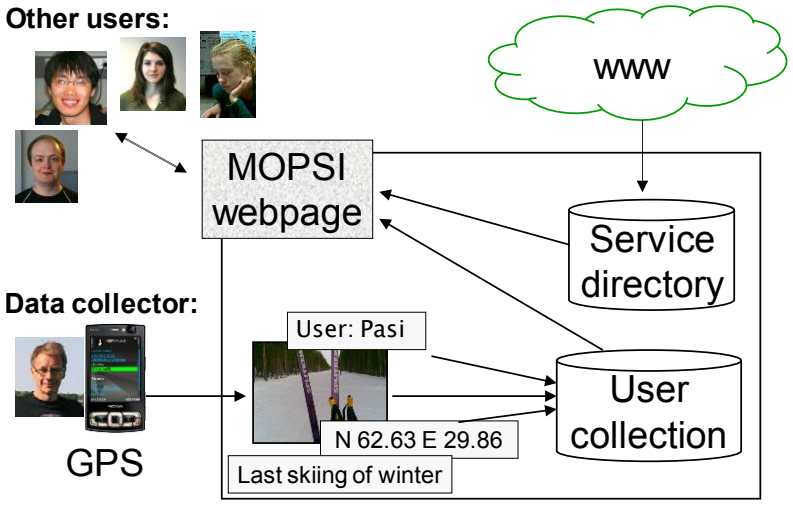

Figure 1-2 Diagram of the MOPSI data collection and services

We study mobile location-based media sharing via the internet in a case study based on the MOPSI service, which is a prototype service for sharing location-based media. The overall structure of the system, outlined in ***Figure 1-2***, consists of two main parts: user collection and service directory.

The main limitation of this kind of ad-hoc information sharing is unawareness of the material of others, especially if the users are not directly linked with each other. The data may be available in the service, but the problem is how to find the *relevant* data from a service with a large number of users. We argue that relevance can be defined by the following aspects:

1. Content of the data
2. Location

3. Time
4. Author and his/her network

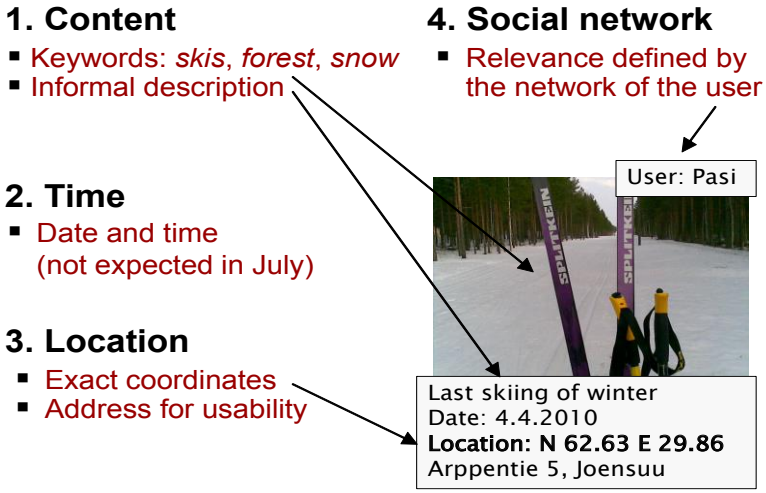

Figure 1-3 Four aspects of relevance in practice

These four aspects are demonstrated in ***Figure 1-3*** by a concrete example, where a person wanted to capture the following scenario. From the photo and its description we can see skis, forest and snow, which relate to wintertime activity. The data also reveals when and where the picture was taken. In 4th April 2010, there were skiing tracks available, which was not self-evident even for citizens of Joensuu. Knowing the proper location was essential. The last piece of information is the identity of the user himself. Strangers may not benefit much of this information, but those who know him and share the same hobby are more likely to find this useful.

1.2.1 Content

Traditionally the relevance is defined by *content* either by user-given keywords or using a predefined format in a database system. This requires a well-designed static database where the service provider models the user behavior beforehand and provides information in the form of a service directory.

On the Internet well-defined attributes are not used, but relevant content can still be found from free text using search engines if the content matches the keywords provided by the user. Tagging of the photos can also be done afterwards, but usually a free-form textual explanation is simpler. It also serves the purpose of social media.

In our application, instead of using manual tagging, we support free-form text description. We implemented queries based on time, location and content for browsing our data on the web. We also implemented a simple recommendation framework based on user location and rating of the photos [WaTF12].

Further analysis of the relevance of content-based image retrieval could be done based on features such as color, texture and shape. Automatic image categorization aims at converting visual content into a set of keywords to describe the content. In [CSLJ09] and [YKSJ09] both visual content and user tagging are jointly applied to recommend the group where a photo should fit best.

1.2.2 Location

Exploiting the l*ocation* of the user has become popular due to the wide availability of GPS positioning in multimedia phones. In case of lacking GPS coverage, positioning can also be provided by the cellular or WiFi network of a mobile phone, or even by using the IP address for a rough estimation of location. Once the location is known, it gives significant additional relevance that can be utilized in several different ways. In our system, location is the key element and it provides additional relevance in the following ways:

1. Browsing data collection on a map
2. Showing the location of other users
3. Tracking the movements of the user
4. Filtering relevant search results for the service directory

Figure 1-4 demonstrates the map view where photos have been clustered and then shown using Google Maps API.

Figure 1-4 Map view of the data collection

Location of users has been visualized in ***Figure 1-5***, using a so-called *smart swap* algorithm [ChZF10] that provides accurate clustering in real-time. For representing the clusters, approaches using icons, grids, Voronoi diagrams, and coloring by the density have been considered in [Delo10]. We use a color bubble attached with the text representing the most recent users in the cluster. The browsing is supported by a zooming operation to get inside bigger clusters.

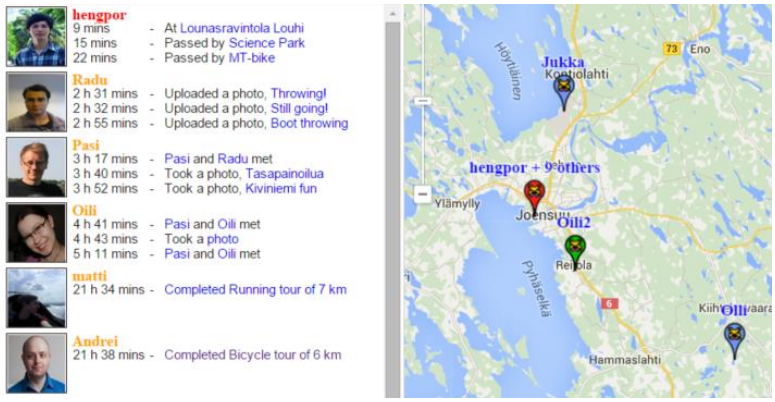

Figure 1-5 Map view of user locations

The collection can also be used as a part of service directory in MOPSI either in mobile phones or on the web, see **Figure 1-6**. Given the location, the user enters a query by using keywords, but instead of providing relevant search results by content alone, results nearby are given if they exist in a local database (green), or found in the user collection (yellow).

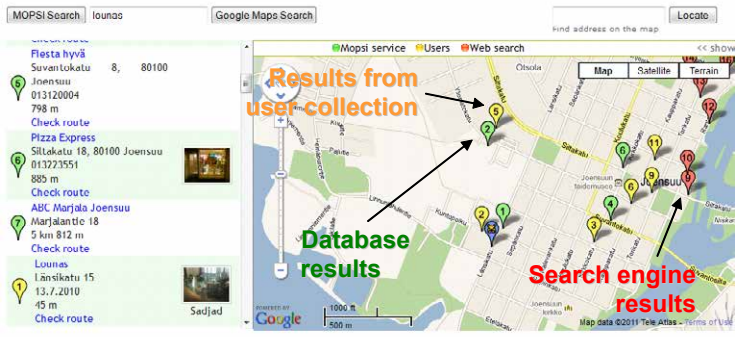

Figure 1-6 Web page interface to the service directory

Additional information (red) is provided by *location-based search* [P2], which is a combination of traditional location-based service and search engine. Following the idea in [HuFi10], our system allows users to transfer search engine results (red) into the service directory (green) by adding proper keywords similarly, and by using photos from the user collection (yellow).

1.2.3 Time

Time can be added to the relevance of the data in several ways. Firstly, the information may be relevant only within a specific time period. A concert or a sports event taking place at a date and time is essential information for the participants. In photo collections, it is also relevant to know when photos were made. In our collection, we utilize this by providing a time line view of the data as shown in ***Figure 1-7***. A similar layout was considered in [SeBD09], with the addition that also links to Wikipedia are supported, to provide more information in addition to the photos.

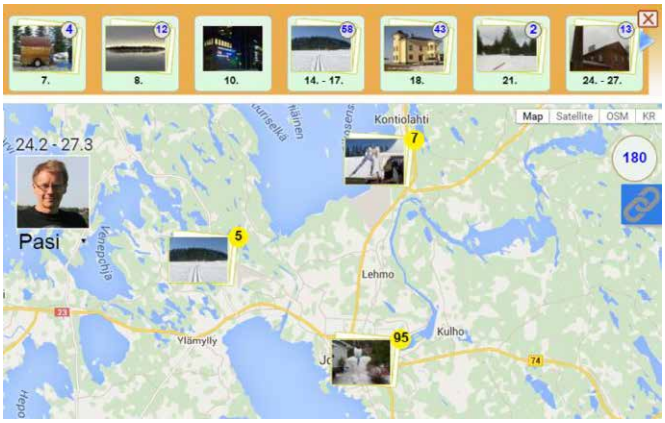

Figure 1-7 Time-line view of the data collection

Secondly, the time and location themselves can be the essential data from an exercise session. For example, the skiing track shown in ***Figure 1-8*** records the length, duration and average speed. This is typical record keeping in the training of a cross-country skier. Although there are specialized GPS sport trackers, the use of our service and mobile phone allows for automatic and real-time sending of the data to the server for user convenience. Moreover, photos can also be taken from the same session by the same device, and presented later jointly with the trajectory of the user as proposed in [PCRC08].

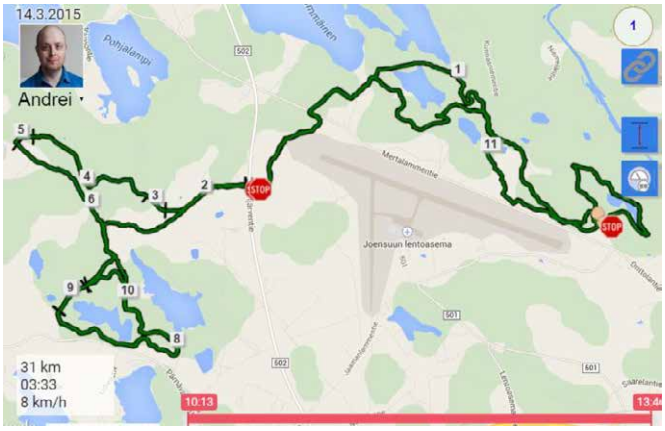

Figure 1-8 Joint time and location for tracking sport activity

In our collection, tracking user's routes is one of the main functions. The web interface also provides navigation from the current location to the location of the search result using Google Maps API based on road maps. An interesting idea for future consideration would be to use the route collection of all users to offer better navigation for pedestrians and hikers instead of the road network that is more suitable for cars [KaKa09].

The third possibility to utilize the time information is to consider the age of the data. The newer the information the more likely it is still valid, as the life expectancy of cafeterias, for example in typical metropolitan areas are often measured in months rather than in years. Moreover, information such as weather condition is needed right here and right now. In ***Figure 1-3***, the skiing conditions are recorded for 4th April, but are hardly relevant for users in July.

1.2.4 Experiments

Next we will provide an overview of the data collected by our users until 25th October 2010. The collection includes many test photos, and the number of users is small, which may somewhat skew the results. Nevertheless, some trends and observations can be seen.

In total, there were 3.589 photos of which most are city views (839), followed by pictures of nature (801) and other people (279). There are also a few pictures of events (90), documents (40) and animals (59). In addition, there are photos that are counted as test photos or failures.

Another point of view is what kind of descriptions has been typed in by the users. Due to the experimental stage, a large percentage of the photos (27%) lack any description. The lack of descriptions is also caused by the difficulty to type by mobile phone, but descriptions can be added later from the web interface.

Among the photos that have some kind of description, a significant share (35%) are only random, some test word (*Symbian_test*), or very generic object descriptions (*Mug*, *Wires*, *Mouse*) indicating test use. In total, 65% of all photos have a

meaningful description. The most documented descriptions are travel photos of places (685), nature (579), general objects (263), architecture (212) people (210), and a few general descriptions of events and animals.

People are often described by their names, or by their roles (*runner*, *floorball player*). Only few are related to place (*Untung / STMIK*), age (*Young Andrei*) or relationship to the person (*my son Amir*).

Events are significantly more often found in the user description than could be concluded by content analysis alone. In our case, events include mostly work-related meetings described by their acronyms (*ecse*, *abi*, *mopsi meeting*, *ubiikki*) but also running competition (*Åland half marathon*) and actions attached with feelings (*quality time in skiing elevator*).

Another difference between content and user descriptions are in travel photos. The location is not easy to recognize from the content but it could be concluded from the positioning data. For example, *Clarke Quay*, *Geger beach*, *Suceava*, *Tahkovuori* and *Aholansaari* are locations whereas the following descriptions include additional details: *Petronas Towers* (building complex), *Heureka* (science center), *Singapore flier* (Ferris wheel) and *Olavin linna* (*castle*). The extreme case is *Musta Pekka mutkan takana* (Black Pete behind the curve) where Black Pete is the name of a particular slope in Tahko skiing resort.

Table 1-1 compares the textual description used in our system with two other photo-sharing sites, Picasa[2] and Flickr[3]. In our system, location is provided automatically without any user interaction, whilst in Picasa or Flickr, location is either manually annotated by users or taken from the photos' meta information, especially if they are taken using mobile phones.

[2] http://picasaweb.google.com

[3] http://www.flickr.com

Table 1-1 Distribution of keywords (tags) used in Picasa and Flickr, in comparison to the user descriptions of the MOPSI collection.

Description	Picasa	Flickr	MOPSI	
			All	Real
Places	---	28%	21%	32%
Events and action	31%	17%	5%	7%
People	6%	7%	6%	10%
Objects	---	5%	8%	12%
Architecture and nature	25%	21%	23%	37%
Animals	---	3%	2%	2%
Other	20%	16%	---	0%
Garbage	19%	2%	35%	---

In Picasa, users provide the location by dragging the photo on Google Map. Keywords and location are thus provided explicitly as two different entities, and consequently, users tend not to type any location related keywords. Flickr has a somewhat more complicated interface based on *Yahoo! Maps*. Only a predefined set of keywords are allowed, which explains the quality of the tags (only 2% garbage).

Despite the automatic positioning in our system, it does not reflect on the distribution of the type of descriptions written. Unlike in Picasa, users still tend to describe the location anyway for travel pictures, probably because the position is not confirmed in the device, but it's retrieved in the background. Overall, the distribution of topics is rather similar to that of Flickr. There are slightly more people and objects described, but these could be just artifacts from the system being in the testing stage.

For the purpose of photo collecting, two mobile applications were developed (Java and Symbian C++). Samples are shown in ***Figure 1-9***. A large number of failures were caused by the Java version, which lacks several important features. Firstly, there is an unavoidable delay from the *click* sound and when the photo is actually taken. People tend to move the camera right after they hear the sound and before the actual picture will be taken. Secondly, auto-focus supported by Symbian helps very much

with picture quality, but it was not available in Java. Other typical failures originated from low quality cameras that do not work well in low illumination. A few faulty pictures were caused by irrecoverable transmission errors.

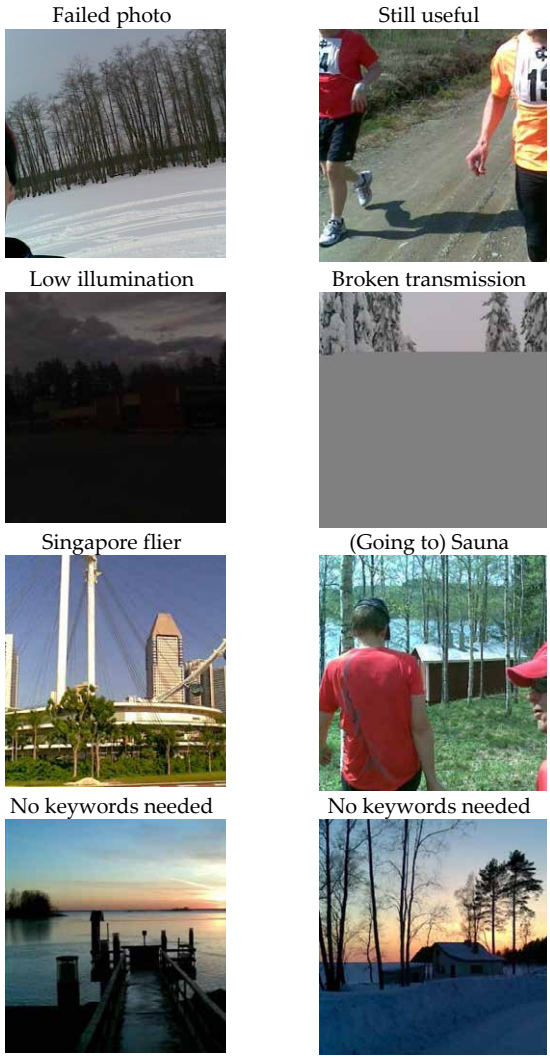

Figure 1-9 The photos in the first row are examples of software problems (click sound), the second row of low illumination and broken transmission problems. The rest are successful photos

1.2.5 Conclusions

We have presented a tool used for collecting user data (mainly photos and routes), serving as a test bench of new ideas, and a prototype service directory. We have discussed how different aspects of relevance appear in location-based data and tested the named aspects in different modes such as search, map views, or timelines. Our tool can be used later for mobile location-based games and as an educational tool for teaching principles of GIS. Although our tool is used for collecting data from users, the same discussions and conclusions on the aspects of relevance can also be applied on location-data that are generated through other means, for example resulting from automated web search or data mining.

2 *Location-Based Web Search*

Location is an important factor to personalize web search. This is because the content of a website has an area of interest that influences its relevance [BRWY11].

The volume of geospatial data is increasing as more and more devices have access to the Internet and positioning technology [PaMP03]. A large part of this multimedia data are nowadays generated with devices that automatically annotate them with location information, but free-form content such as websites do not implicitly contain any geographical information. Mobile search engines allow users to find information anytime, anywhere and the search performance is influenced by the type of mobile device and the user's context, which is influenced by aspects such as location, profile, previous activity, time of year or social network [LiRG10].

Location-based services such as Fonecta[4], Google Maps[5] and Nokia Ovi Services[6] emerged very fast in our everyday lives via mobile phones and other consumer electronics. Their main limitation, however, is that they are fully or partially based on databases where the entries must be explicitly geo-referenced beforehand when added. *Search engines*, on the other hand, are efficient in finding information from the Internet without any prior knowledge or explicit search structure. Their limitation is that the location of the user is not yet well utilized in the current solutions. This is because the information on web pages is rarely attached to the location for which it would be relevant.

[4] http://www.fonecta.fi

[5] http:/maps.google.com

[6] http://www.ovi.com/services/

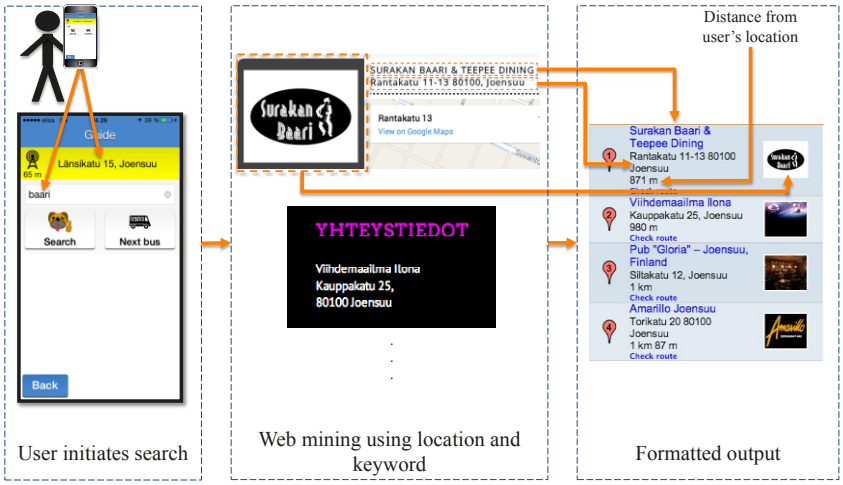

Figure 2-1 Web mining using location and keyword

Location-based search aims at finding a business or place of interest around a specific geographical location. This is supported by search engines that support geographical preferences [MCSL05]; the relevance of a search result depends on the distance between the user-specified location and the location of the service [YoTM01]. Location-based search changes the search from web-oriented to service-oriented, which makes it a more challenging task due to two reasons. First, it is not enough just to find a relevant web page for the user as in traditional web search. Instead, we also need to detect the location that the web page is relevant to. Second, we need to extract information from the web page. This is either a simple summary of the content (such as title and image), but in case of service directories, we need to extract the part belonging to that particular service. It requires both the identification of geographical data and automatic information extraction from web pages. A simplified workflow of a location-based application that also applies to our system is shown in ***Figure 2-1***.

Locations are embedded explicitly as geographical coordinates (usually latitude and longitude), both in source code as HyperText Markup Language (HTML) meta tags named *geo-*

tags[7] and in the content of the web pages as plain text. Locations are embedded implicitly as *geographical references* that are found in the text content of the web pages in many ways, such as: postal addresses, place names, descriptions in natural language and driving directions. Identifying geographical references and associating a web page with one or multiple locations is a process called *geo-referencing*. A particular case of geo-referencing is *geo-tagging*, which is the operation of assigning geographical coordinate metadata to multimedia such as photos, videos and websites.

Very few web pages are using explicit localization. A world-wide study does not exist, but according to [Väns04] less than 0.1% of Finnish websites were using geo-tags in 2004. Furthermore, less than 1% of the websites related to the German city of Oldenburg were using explicit localization in 2008 [AhBo08a] and 7% of the service websites from Finland collected in MOPSI until May 2015 [P4]. Therefore, the main method of geo-referencing web pages is to detect the implicit locations from their content. According to [Mccu01], including postal addresses is the most common method of implicit localization, especially for the pages that describe commercial services.

In this section, we propose an alternative solution based on web search and ad-hoc geo-referencing. We aim at combining the benefits of web search and traditional location-based services exploiting the location. We define a location-based search engine as being a web search engine in which the geographical location is an additional relevance criterion. The general idea behind our solution is the first implementation of the idea originally outlined in [HaFM02] and improves most of its technical aspects. Here we describe the technical solutions for implementing the system on multimedia mobile phones and provide experimental comparison of its search capability, in comparison to existing location-based service solutions such as Google Maps and Yellow Pages.

[7] http://www.w3.org/2003/01/geo/

2.1 LOCATION-BASED SEARCH APPLICATIONS

The first methods of assigning locations to web resources, such as [BCGG99] and [WaAm03], rely on identifying the host locations, which are the location of the owner or the administrator of the website. These methods assign a single location for each website by querying its *Whois*[8] records for the address and telephone number of the network administrator. This method is expanded in [Mccu01] by additionally using hyperlinks, meta tags and postal addresses as sources for location information.

Most of the websites are not geo-tagged by default and their content cannot be directly used by location-aware services. Web pages are designed to be browsed by humans and contain geographical references that are complex, informal, diverse, ambiguous and difficult to be processed by a computer [ShBa11]. However, we can adopt an unsupervised process of extracting locations from web pages. According to [HuLR05], several strategies can be used for geographic reference extraction: text matching using gazetteers, rule-based linguistic analysis, text matching based on regular expressions, identification of host locations and reading geographic meta-tags. Our application uses text matching based on gazetteers and regular expressions.

According to [WXWL05], there are three types of locations that can be inferred from web pages: provider location (where the owner of the page is), content location (where the content is pointing to) and serving location (the area for which the web page is relevant). Provider location is detected using a set of heuristic rules such as referred frequency, URL levels and spatial positions of address strings in the web page. Content locations are calculated by extracting all geographical references with the use of probabilities that measure the reliability of each source. The probabilities are based on the measures of power and spread of a geographical reference [DiGS00] and use a location hierarchy *country-state-city* in the form of a geographic tree. Serving location

[8] http://www.whois.net

is found in a similar way, but it additionally uses links between pages and user visits logs. Our application is focused on detecting the content location by extracting geographical references.

Postal addresses are the most common way geographical references are found in web pages [GoWK07]. They are converted into locations by services that provide *geo-coding*, which is the process of finding geographical coordinates (usually latitude and longitude) from other types of location data, such as street addresses or postal codes. Our application detects addresses using free geo-coding services. We use OpenStreetMap services and build a geo-coded database for Finland with publicly available data. According to [FLMN10], because of occasional service unavailability and data accuracy, free geo-coding services such as OpenStreetMap or Geonames [9] are best suited for applications that require almost accurate geo-tagging, whilst systems that deal with public health or vital services require higher quality data.

One of the biggest challenges that arise when identifying locations from a web page is the ambiguity of terms, which can be between locations with the same name (known as *GEO/GEO ambiguity*) or between location names and non-geographical entities (known as *GEO/NON-GEO ambiguity*). Both types of ambiguity can be resolved using heuristic rules, but the algorithm [ZJLY12] additionally attempts to resolve the GEO/GEO ambiguity using an algorithm similar to Google's Page Rank [PBMW99]. In our case, ambiguity is a smaller problem because we detect postal addresses, which have a low level of ambiguity when they contain accurate elements such as postal code.

There has been several works reported in the literature for location-aware search.

A location-based search engine for the Singapore area [Tsai11] is able to search for locations by using filters on area names, building names, landmark types, business names and business

[9] http://www.geonames.org

categories. It uses the location of the user along with search filters to select items from a catalogue of businesses and landmarks.

A personalized mobile search engine enhanced with capturing users' preferences in the form of click-through data is proposed in [LeLL13]. The users' preferences are captured in the form of concepts, which are modeled as ontologies and separated into location concepts and content concepts. The search engine also considers users' GPS location and uses content and location entropies to balance the content and location concepts. The locations from documents are detected using a predefined ontology that uses city, province, region and country names.

A location extraction method that receives websites as input and equips them with location tags, being able to extract location with a precision up to street level is outlined in [HeMS13]. Words are extracted from the web page and checked against free gazetteers (Geonames and OpenStreetMap) using Aho-Corasick string matching algorithm [AhCo75]. The validation and disambiguation of the locations is done by detecting their context with the use of the other geographical references from the text. The method outperforms a commercial solution (Yahoo! Placemaker), but it correctly detects just 60% of locations. The authors demonstrate the applicability of their method by describing three practical applications derived from their work: location-aware Web surfing through a mobile device, browsing by using nearby tags, and location tagging through social networks.

A system that is capable of handling geographical queries of the triplet of <theme> <spatial relationship> <location> is described in [PCJA07]. It handles spatial relationships such as *inside*, *near*, *north--of*, *south-of* and geo-references, stores and indexes web pages using both pure text and spatial indexes. Using both types of indexes enables a full set of geographical query operators, graphical query formulation and the ranking of results according to conceptual as well as spatial criteria. Geographical ontologies are used for query expansion and for disambiguating the queries and the extracted locations. The

system relies on web crawling, pre-processed indexes and combines textual and geographical relevance. Locations are detected using a gazetteer lookup approach, which is enhanced with context rules and additional name lists used for filtering.

A system that leverages on contextual information, which can be used in the named entity recognition and disambiguation steps is described in [QXFX10]. A set of location evidence is built, updated and used to provide geographical contextual evidence.

A method to identify address data by combining patterns and gazetteers from free sources such as OpenStreetMap is used in [SMRS13] to identify companies from web pages. The web pages are pre-processed by removing HTML tags, extracting text, line splitting, tokenizing and part-of-speech tagging. The single attributes *postal codes*, *city names*, *street names*, *street numbers*, and *company names* are identified on the pre-processed data using regular expressions and heuristic rules. The attributes are then aggregated starting with the company name.

A method for extracting postal addresses and associated information using sequence labeling algorithm is introduced in [ChLi10]. Unlike most of the existing methods, postal addresses are not detected by using gazetteers. Instead, addresses are detected by pre-processing data with a named entity recognition tool, extracting features from text and training models using support vector machines and conditional random fields. Pattern mining is applied to identify the boundaries of address blocks and to extract the associated information for each detected address. The associated information is defined as information that refers to the detected addresses and allows better comprehension.

A knowledge-based web-mining tool that adopts a geospatial ontology, a rule based screening algorithm and inductive learning for automated location retrieval is described in [LGCZ12]. Address detection is customized to discover the locations of emergency service facilities; other detected addresses are discarded.

A method that analyses the DOM tree of a web page is used to extract product data from company websites [DoHu12]. The leaf nodes of the DOM tree are analyzed and used to generate semantic information vectors for the other nodes, which in turn are used to generate a maximum repeating semantic vector pattern. The generated pattern is used to detect product data regions and to build product templates, which are used along with a semantic tree matching technique to identify product information.

A vision-based approach of extracting data records from web pages is proposed in [LiMM10]. Instead of using HTML structure and DOM trees, the method uses the visual block trees generated using the algorithm described in [CYWM03]. The visual block tree is primarily based on the visual features that humans can capture from web pages, using the information from the page layout and attributes such as fonts and background color. The data records are extracted based on the visual block tree and the visual features of its elements, which are rectangular data blocks. These data blocks are filtered, clustered and regrouped to identify data records.

2.2 CONTRIBUTION

In this section, we propose a method for extracting locations from web pages and complement it by extracting a title and a representative image for each search result. Our goal is to integrate processes of geo-referencing, geo-coding and geo-tagging into a unified location-based system that is able to provide relevant information. This information has to be close to the user's location, related to the keywords provided by the user and extracted from the content of websites.

A schematic diagram of our proposed system is shown in ***Figure 2-3***. The proposed method is implemented in the framework of MOPSI that is a research project of location-based services [FKTS10]. Besides location-based search, MOPSI offers

tools for collecting, processing and displaying location-based data, such as photos or trajectories, along with social media integration.

The contribution of this section can be summarized as follows:

We propose a location-based system that uses a meta-search approach similar to [LeLL13]. We use the results of a search engine that is not location-aware and post-process them by extracting the locations from the provided websites. The meta-search approach has the advantage of allowing us to find relevant websites without crawling, indexing and storing websites, but it makes our system dependent on the relevance of other search engines and vulnerable to changes in the external search engines we use.

We extract associated information, as in [ChLi10], to bring better comprehension of the search results, in our case the name of the services or places the locations are referring to and a representative photo. The associated information is extracted by using rule-based heuristics and the DOM tree nodes that contain location information.

We download and parse the HTML source of the provided web pages and we use its DOM tree representation, similarly to [DoHu12] and unlike many methods that are concentrating just on plain text extraction.

Our approach is different than [QXFX10] because we rely on gazetteers in the address detection step, but we are also using contextual information in order to detect the entities that are related to the detected locations.

We find locations by detecting postal addresses. The proposed method in [DoHu12], using DOM, is effective and it can be applied also for detecting locations and associated information, but is limited to the websites that contain a list of items with a clear and repeating pattern. The address elements are identified individually and then aggregated in order to build an address candidate, in a method similar to [SMRS13]. The difference is that we start the aggregation with the street name and we detect the additional information in the next stages. We identify address

elements using the gazetteer approach and regular expressions. Our process is service-oriented, so we need accurate locations, not just areas as in other works, and we consider the location as postal address. Our approach is lightweight as it does not require training, but it is dependent on the quality of the gazetteer data and the addresses on the web pages.

We validate the addresses by using a gazetteer and then find the respective coordinates. In this case, disambiguation is not a problem because we aggregate elements that form a unique location, for example street numbers and postal codes.

We do not associate a single location to a single web page, but we build an ordered search result, which we rank by distance from the user's location. The search results are general and they are not limited to a theme or a type, such as products or companies.

Our search engine does not use pre-collected databases of services and relies on automatically detecting locations and extracting data in real time from web pages. Compared to [LGCZ12], we aim at a broader scope for our application and we do not limit its use to a certain type of service.

Our system is flexible and can detect locations from any set of web pages, not just the results of an external search engine. The implementation is not limited to specific geographical areas, although it is dependent on the accuracy of the gazetteer data we use. For this purpose, we use OpenStreetMap gazetteer data, which is available for most countries.

In respect to existing commercial services such as Google Maps, Bing Local[10], Yahoo Local[11] and Yellow Pages, our goal is the same: provide location-relevant information to the user. However, these applications are mainly based on commercial databases, user input and pre-collected data resulted from web crawling, and only exploit the results of real-time web search to a

[10] http://www.bing.com/local/

[11] http://local.yahoo.com/

limited extent. A location-based search engine is an alternative approach for information retrieval to traditional location-based services based on fixed databases. It aims at utilizing the location of the user, but without being restricted to any fixed location-based service.

2.3 MOPSI PROTOTYPE

Our location-based system [P4] takes as input the user's context, which is location, city and search keyword and outputs an ordered list of search results that contains the following information: rank, title, URL, location, address, representative image and distance from the user's location (see ***Figure 2-1***). The web interface of our system in shown in ***Figure 2-2***.

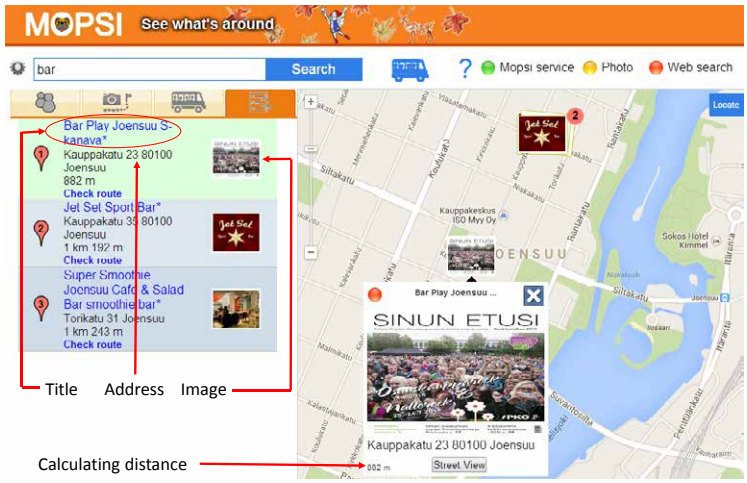

Figure 2-2 The web interface of the MOPSI search engine

The search workflow is detailed in ***Figure 2-3*** and starts by finding websites that are relevant to the location and the search keyword provided by the user. This is done by the *website provider*, which queries a conventional search engine with the <keyword, city> phrase and outputs a list of websites. A part of the *data extraction* module, the *web page parser*, downloads the web pages detected by the website provider and outputs their HTML source along with their DOM tree representation. The *address detector*

module then marks the nodes in the DOM tree that contain addresses and outputs a list of address candidates. The marked nodes are used by the *title and image extraction* module to detect a representative image and a title for each detected address. The address candidates are validated by the *address validator*, which uses our gazetteer based on OpenStreetMap. Finally, we rank the locations by distance and aggregate all the detected attributes as search results, which we display to the user as a ranked list.

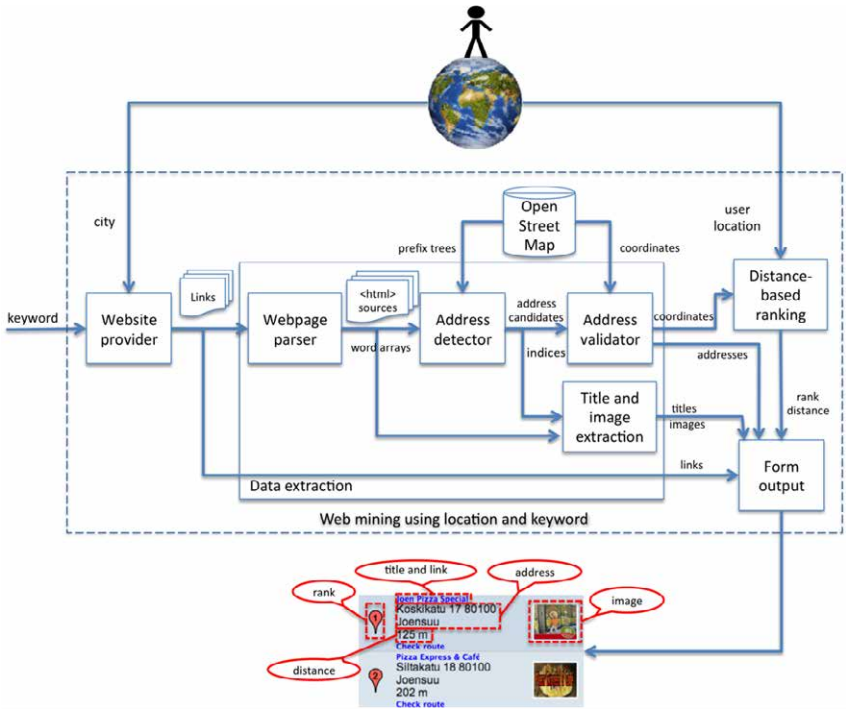

Figure 2-3 Location-based search workflow

2.4 LOCATION-BASED SEARCH MODULES

The *website provider* takes the user's context as input (city and keyword) and provides a set of URLs that are relevant to that context. The URLs are later used in the data extraction processes.

Services such as BOSS API by Yahoo![12] or Custom Search by Google[13] allow third parties to build search products by using the infrastructure of their search engine. The website provider uses those services to perform a conventional text search using the <keyword city> query. The website provider relies on the relevance of the results of the external search engine. It uses the keyword and the city provided by the user and it does not expand the query in any other way. In order to reduce the time of the search, we just use the first 10 results of the query to build the list of websites that is the output of this module.

The *data extraction* module extracts location-based data (title, URL, location, address and representative image) from the collection of web pages detected by the *website provider*. It includes the following sub-modules: *web page parser*, *address detector*, *address validator* and *title and image extractor*.

The *web page parser* uses the Document Object Model representation to generate a tree structure for each web page it downloads. The DOM tree of the web page is used in latter stages for detecting locations and location-related information such as service name and representative image.

The *address detector* searches for postal addresses on the web page using a text matching algorithm based on street-name prefix trees [P3]. It uses the text nodes of the DOM tree of the web page to identify the following address elements: street name, street number, postal code and city. Address candidates are constructed by aggregating address elements that are close to each other in the text of the web page. The prefix trees are constructed on demand for each city using our own gazetteer for Finland, see [TaFM09], and OpenStreetMap for the rest of the world. The address detector outputs a list of address candidates and marks the nodes that hold location information.

The *address validator* uses OpenStreetMap geo-coding services to validate the addresses detected at the previous steps and

[12] http://developer.yahoo.com/boss

[13] https://developers.google.com/custom-search

converts them into geographical coordinates. The coordinates are then used for displaying the results on the map and for computing the distance from the user's location. The postal address candidates that are not validated by the geo-coding services are discarded.

The *title and image extractor* identifies the title and relevant image associated with the detected locations. It uses the DOM tree and searches for text and images in the sub-trees of the nodes that contain location information. The output is a list of geo-referenced entities that contains the information described in ***Figure 2-3***.

Finally, items are sorted using *distance-based ranking* and all the information is assembled by the *form output* as a list of search results.

2.5 WEB PAGE PARSING

HTML documents are considered to be semi-structured data, which are neither raw nor strictly typed [Abit97]. HTML documents do not conform to a formal data model in the way that structured data such as a database do. They are not structured because they do not have a fixed schema and because their elements typically hold information solely for rendering. They are not completely unstructured because the HTML tags and their tree structure can be used to guide data extraction.

An HTML document can have a DOM representation, which is a platform- and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure and style of documents[14]. Therefore, an HTML document can be represented as a tree made up of parent-child relationships between the HTML elements. A parent can have one or many child nodes and the <html> tag is the root element of the tree. ***Figure 2-4*** shows a simplified example of an HTML

[14] http://www.w3.org/DOM

page, where we only display the sub-tree that contains the location information. After we detect the text nodes that contain postal addresses, we use the tree structure of the HTML document to detect the title of the service, see section 2.7.2.

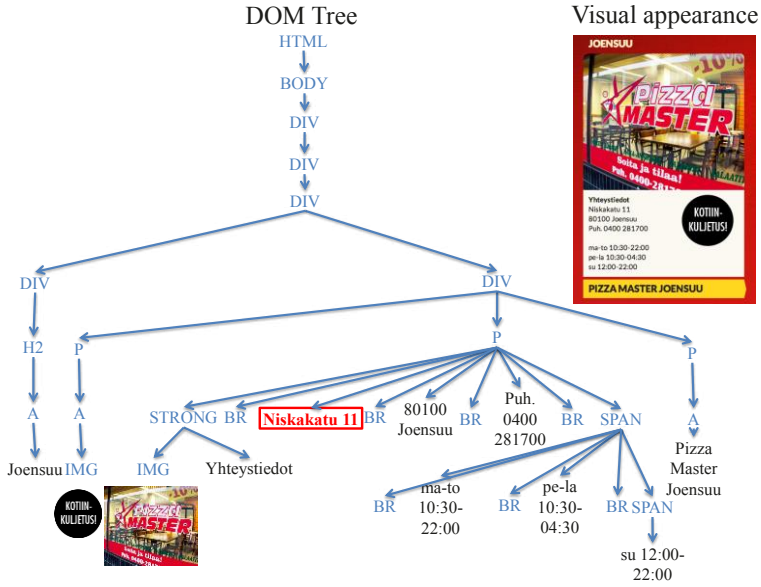

Figure 2-4 Part of a web page that contains location information: visual appearance and DOM tree

2.6 ADDRESS DETECTION USING STREET-NAME PREFIX TREES

One of the most important problems in our location-based application is how to extract and validate the geographical locations from free-form or semi-structured text.

In most languages, the street-name expression of an address is one out of several common terms. In English, it usually ends on *street*, *way*, *drive*, *road*, and in Finnish on a suffix such as *-katu*, *-kuja*, *-tie*. A simple heuristic-based method described in [FKTS10], detects street names using regular expressions with predefined endings (suffixes) and it works reasonably well for Finnish addresses. However, not all street names follow a predefined pattern, and names that have a different suffix or structure would not be detected.

A common way to detect addresses from free form text is to build and train a classifier as in [ViNa05]. However, customizing the classifier to other languages and countries takes considerable work as new ground truth tagged text corpus must be created by hand.

Another way of matching the potential address strings is to use *brute force* by comparing each word in the document to the entries in an address database. However, this can be rather inefficient if the database is large.

Detecting postal addresses and street names is a *Named Entity Recognition* problem and is usually done with the help of gazetteers. Named entity recognition without gazetteers is discussed however in [MiMG99], but it turns out to have bad results in detecting locations and addresses.

Nevertheless, most of the applications that identify postal addresses are using gazetteers. For example, the Web-a-Where system [AHSS04] uses a gazetteer and a three-step process: spotting, disambiguation and focus determination. Our address detection algorithm shares the first two steps (spotting and disambiguation), but it aims to find individual locations and services on web pages, while Web-a-Where assigns a geographical focus to a web page.

An ontology-based approach that extracts geographic knowledge is presented in [BLMD07]. The address is divided into three parts: basic address (street and building number), complement (optional, may be neighborhood name) and location identifiers (phone number, postal code, city name) and a spatial index called geo-index are built for each page. Address detection is done using the gazetteer as described in [SDBD05] for geo-parsing and geo-coding. It relies on a set of rules and patterns implemented as regular expressions using four elements: *basic address*, *postal code*, *phone number* and *city/state*. Our gazetteer also uses an ontology based on address elements, but our rules are more flexible because our address candidates do not impose any order in the address elements.

Another ontology-based method is described in [CaWJ05]. Postal addresses are detected using conceptual information retrieval combined with graph matching. Concepts, which are knowledge and address elements, are identified and linked in graphs by using semantic relations. Address detection is done by matching the graphs from a web page with template graphs that represent addresses. Geographical concepts are identified using a concept set that is a gazetteer.

A syntactic approach to postal address detection is described in [CQXW05]. It consists of two steps: vision-based text segmentation and syntactic pattern recognition. In the text segmentation step, the application analyzes html tags and identifies two types of blocks: cue blocks (for the purpose of indications, annotation, and explanation) and body blocks (main text body content). Postal address detection relies on calculating the confidence of the identified blocks, which in turn is based on tokenization of the words as city names, state names, street and organization suffixes. Our approach is different because we rely mainly on street name detection.

A graph-ranking algorithm for assigning a geographic scope to a web-page is proposed in [SMCA06]. Address detection is aided by a geo-ontology knowledge base, which uses a set of rules, relationships and heuristics.

A method of ranking web search results using geographical information is described in [LeLM07]. Address detection is done by using a regular expression to detect typical address elements. Once a pattern is detected, the street name and city name are validated using a gazetteer. Our method is different because by aggregating address elements, we consider more patterns for postal addresses, and because we use faster methods for gazetteer matching.

In [AhBo08b], describes a geo-parser that identifies address level locations. Instead of relying on metadata or other structured annotation, the geo-parser uses a gazetteer database. The database used by the geo-parser contains postal codes, city names and street names. Every city-postal code combination is also used for validation. The address detection assumes that the address

blocks have a certain structure, and that there are certain dependencies between the address elements. We also utilize the idea of identifying a number of address elements and validating the address by geo-coding it.

2.6.1 Proposed Method

We propose a rule-based text-matching solution that detects address-based locations using gazetteer and street-name prefix trees created from the gazetteer. Instead of extracting the raw text from a web page, we are using the DOM tree representation and navigate through its text nodes in order to detect postal addresses. This way we can mark the nodes that contain location information and use them to detect additional information such as title and representative image. For each node of the DOM tree, we extract the text of its sub-tree and use it for address detection. This allows us to find addresses that are spreading through several nodes (for example if one element is bold) or tables.

```
nodes = GetDOMTree(url)
cities = RetrieveCities(nodes)
FOREACH city IN cities
  prefixTree = RetrieveStreetPrefixTree(city)
  FOREACH node IN nodes
    AddressDetection(node, prefixTree)

AddressDetection(node, prefixTree)
  words = ExtractSubtreeText(node)
  FOREACH word in words DO
    IF prefixTree CONTAINS word THEN
      Search for number, postal code, city name near word using regular expressions
      IF number, postal code or city name found THEN
        Aggregate found elements into address candidate
        Get coordinates of address candidate using gazetteer
        IF coordinates found THEN
          Add address candidate to address list
          Mark node as containing address information
```

Figure 2-5 Pseudocode for address detection

Our address detection algorithm (see ***Figure 2-5***) starts by identifying street names. For each detected street name, we analyze its neighboring text and identify other potential postal address elements: street number, postal code and city name. We aggregate the detected elements into address candidates, which are then validated by our gazetteer. An address candidate needs to contain at least the following elements: street name, a street number and a postal code or a city name.

The main problem in address detection is ambiguity: we might find sections of a web page that have the same structure as an address, but do not exist in reality. We therefore validate all hypothesized addresses and discard the false detections by using a digital gazetteer, which is a geospatial dictionary of geographic names [HiFZ99]. As a side-product, the validation process provides the geo-coding, i.e. converts the given address to a pair of coordinates, which we use to plot the results on a map or to calculate the distance to the user's location.

In our approach, no ground truth tagging is needed. We only need gazetteers and simple rules on how the street name appears in relation to other address elements. Efficient use of the gazetteer is possible because we know the user's current location and its interest area consists only of those services that are close to him. Therefore, we build a fast access structure to that partial gazetteer, which is a prefix tree. We construct a prefix from all street names in a given municipality and use just the municipalities that are in the proximity of the user's location.

For testing purposes, we have also implemented a heuristic method of postal address detection without a gazetteer, which is much simpler and exploits structural characteristics of postal addresses, but its applicability is limited, as we demonstrate in our experiments section. The proposed solution is faster and more accurate than the heuristic solution alone and much faster than the brute force.

The accuracy of our address detection approach may vary, as each country has its own standards on postal addresses that can

tolerate such variations as abbreviation, different order of elements and missing elements. For example, in Finland it is common that the address block has a <*street-name, street number, postal code, neighborhood, municipality*> structure (see **Figure 2-6**). In Finnish addresses the street name and street type are concatenated, with street name being the prefix and street type being the suffix. Examples of street names are: *Kauppa-* (market), *Alexanterin-* (Alexander's) or *Kaisla-* (Reed) and types: *-katu* (street), *-tie* (road), *-polku* (way) or *–kuja* (alley). Finnish addresses have a fixed order, but neighborhood and postal code are optional. In Singapore the <*street number, street name, street type, postal code, municipality*> structure is more common, but variations exist, for example, a street name can be written using abbreviations such as Ave instead of Avenue, which is rarer in Finnish addresses, or street number can also be written after the street name.

Kaislakatu 8, 80130, Kanervala, Joensuu, Finland
Torikatu 25, 80100 Joensuu, Finland
Parppeintie 6, 82900 Ilomantsi, Finland
Aleksanterinkatu 25, 15140 Lahti, Finland
East Coast Park Service Road 1, Singapore
290 Orchard Rd., 238859, Singapore
22 Orange Grove Road, 258350, Singapore
6 Bayfront Ave, Singapore

Figure 2-6 Example of Finnish and Singaporean addresses

Although the websites we use to test our methods are from Finnish and Singaporean services, our address detection algorithm is not tailored for a specific country or language, although it only detects addresses that follow a structure similar to Western European addresses. Our method does not use a predefined order of address elements, therefore it tolerates variations in the order of elements, as we search for elements both before and after the detected street names, but it could produce false positives since it does not consider any semantic relationship between elements. The support of abbreviations

depends on the data in the gazetteer, which needs to have a separate entry for each possible abbreviation and is difficult to generate and maintain. Furthermore, the text matching method we use supports only exact matching and it does not detect wrongly spelled street names or municipalities. This should be improved in future work.

2.6.2 Gazetteer Database

2.6.2.1 Usage and Common Operations

Location-based applications that use geo-referencing require a data structure or a database that connects any given address to its exact coordinates. Such database is called a gazetteer, which is a geospatial dictionary of geographic names having the following minimum components: geographic names, geographic locations represented by coordinates and type designation [HiFZ99].

In our location-based application, we use a *gazetteer database* to store addresses in the form of <street, number, municipality> along with their corresponding geographical coordinates (latitude and longitude) for Finland and OpenStreetMap for the rest of the world. The gazetteer is used every time an address is converted into a location or when the postal address that is the closest to a known location is required. Furthermore, both OpenStreetMap and our gazetteer database are used for creating the arrays of prefix trees used in the detection of postal addresses on web pages. To speed up the location-based search, the gazetteer database has to be optimized for easy and fast access.

In our location-based search scenario, the following operations are needed:

1. Find all the municipalities within a bounding box

This query is performed when a search is initiated and determines the user's interest area. This is usually defined as a square bounding box with a fixed length and it can intersect only one or several municipalities.

In the first case (***Figure 2-7***, left) the bounding box intersects only one municipality because the provided location is near the

center. In the second case (***Figure 2-7***, right), the user's location is near the border between the two municipalities and the search engine needs addresses and location points from both municipalities.

This operation is performed once per search and its running time is not critical.

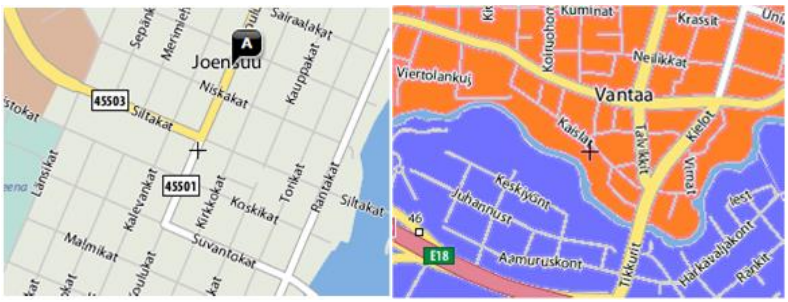

Figure 2-7 The bounding box intersects one municipality (left. Joensuu) or two municipalities (right, orange – Vantaa, blue – Helsinki)

2. Find all the street names from a municipality

In order to detect postal addresses in the selected municipalities, a list of street names has to be provided for each municipality. Instead of using all the street names in our database, we limit the search to the municipalities that are close to the user's location. For fast access, street names are stored in prefix trees.

3. Convert the detected addresses into coordinates

This operation is called geo-coding and it is performed for each detected address. Coordinate conversion is used to calculate the distance from the user's location and to validate the detected addresses. If no coordinates are found, then the address is discarded. This is one of the most time critical operations because it can be performed tens or hundreds of times per search, depending of the number of addresses found.

4. Converting coordinates into addresses

This operation is called reverse geo-coding and is used for determining the user's address.

2.6.2.2 Implementation and results

Our gazetteer uses a database that contains geographical coordinates attached to address strings for Finland. For the rest of the world, we are using the Nominatim[15] project. Nominatim provides an interface to query OpenStreetMap data and has support for the four common operations we are using.

The speed of our gazetteer can be increased by using a database management system that supports Open Geographical Information System (OpenGIS) specifications, using database indexing or data types and functions that implement OpenGIS standards. Such database management systems include MySQL with spatial extensions, PostgreSQL with PostGIS or Oracle Spatial.

We implemented a postal address database that stores all the addresses in Finland using a MySQL5 database management system. For the design of the database the following factors were considered: the use of MySQL spatial extensions or common data types and the use of database indexing.

For testing purposes, a postal address database of the North Karelia region was created. ***Table 2-1*** shows the database sizes for the considered solutions.

Results show that indexing dramatically increases the database size with more than 90%, whilst using spatial extensions also increases the storage size with more than 12%.

Table 2-1 Database sizes for North Karelia region

Spatial extensions	Indexing	Data size (MB)	Index Size (MB)	Database Size (MB)
No	No	22.7	-	22.7
No	Yes	26.7	25.6	52.4
Yes	No	29.3	-	29.3
Yes	Yes	33.3	48.0	81.3

We tested the query execution times using a benchmark application that randomly chose 500 location points within our

[15] https://nominatim.openstreetmap.org/

region and tested the most common operations described in the previous section. We logged the execution times and we calculated the total time for query execution for the proposed testing scenario using the following formula:

$$T_{total} = T_{query1} + n_1 T_{query2} + n_1 n_2 T_{query3} + T_{query4} \qquad (2.1)$$

Above, n_1 represents the average number of municipalities returned by *query1* and n_2 represents the average number of search results.

The values of n_1=1.7 and n_2=64 were determined experimentally and used in the calculation of average query execution times, see ***Table 2-2***.

Table 2-2 Average query execution times

Spatial extensions	No	No	Yes	Yes
Indexing	No	Yes	No	Yes
Query 1 (s)	886	898	3956	176
Query 2 (s)	715	953	1091	173
Query 3 (s)	670	14	674	13
Query 4 (s)	887	909	3866	215
Total time (s)	75	5	83	2

Results show that the non-indexed solutions are at least 15 times slower comparing to the indexed ones, therefore using a non-indexed database is not justified. Using spatial extensions makes queries run significantly faster on the indexed solutions (at least twice as fast) and slower on the non-indexed solutions (1.09 times slower).

The most efficient solutions for the database are also the most storage-costly solutions. However, in our application, the execution time is more important that storage space. The most time-efficient solution, which is using spatial extensions and indexing, turned out to be the most storage-costly one.

2.6.3 Street Name Detection

Street name detection is the starting point of our address detection algorithm and it can be done either with or without the use of a gazetteer. The methods that do not use gazetteer usually assume that a street name has a certain structure, whilst the methods which use a gazetteer rely on fast word matching. For comparison, we implemented both approaches: a *heuristic method* that does not use a gazetteer and two text matching methods that use data extracted from a street name database.

2.6.3.1 Heuristic Method

Heuristic methods rely on regular expression matching. The structure of most addresses has certain particularities. For example, street names can start with the same prefix or end with the same suffix and they can be preceded or succeeded by standard words or numbers

According to our experiments, this approach has good results for Finnish street names because most of them end in words such as *katu* (street), *tie* (road), *kuja* (lane) or *polku* (path). The heuristic method has the advantage of not needing any predefined data structures to store the street names and it is reasonably fast.

Heuristic methods have the disadvantage of needing to be tailored for every country and language because of the various ways an address block can be constructed.

2.6.3.2 Brute-Force Matching Using Street-Name Arrays

A brute-force text matching method checks every word on a web page against a street name database. We use an optimized brute-force solution that checks the word against all street names that are close to the user's location, for example in the municipality where the user is located.

We use arrays of street names that are created beforehand from the gazetteer. Each array stores all the street names in a municipality. The search is done using language-specific functions and since our search engine is written using PHP scripts, we use the *array_search* and *in_array* functions.

2.6.3.3 Text Matching Using Street-Name Prefix Trees

We store street names into prefix trees, which are created beforehand using the data in our gazetteer. In general, the postal addresses are not unique and the same street name can be found in many cities. A prefix tree is therefore built for each municipality. Because the user's location is known beforehand, we limit the search to its area of interest and we load the prefix trees for the municipalities in that area.

Table 2-3 Gazetteer statistics

	Finland	**Singapore**
Number of municipalities	410	1
Total number of street names	92 572	573
Number of streets per municipality	474	573
Average street name length	11.6	6.1
Total size (MB)	74.4	0.18

Our research project is based in Finland and we additionally used Singaporean street data, as the addresses have a higher degree of variation than Finnish addresses, are written in English and were easily available to us when the experiments were performed. Statistical data about both gazetteers are detailed in ***Table 2-3***.

2.6.3.4 Street-Name Prefix Tree

The prefix tree (or *trie*) is a fast ordered tree data structure used for retrieval [NaRa02]. The prefix tree stores a collection of strings, indexed from the beginning of a word (i.e. prefix). The root node represents an empty string and its children store the first letter of each of the indexed strings, their children store the second letter and so on. The same principle is applied at every level of the tree so that the internal nodes describe all the sub-strings (prefix) of a particular string. The recursive version of the algorithm is presented in **Figure 2-8**.

```
ConstructTrie(streetnames)
  Create empty node root
  FOR i = 1 TO count(streetnames) DO
     AddString(root, streetnames[i], i);

AddString(node, string, index)
  IF (length(string) > 0) THEN
    IF (string[0] is not the key of a child of node)
    THEN
      Create new node child with the value string[0]
    ELSE
      Set child as the child of node with the key string[0]
      AddString(child, substring(string, 1), index)
  ELSE //node is a terminal node
    node.index = index;
```

Figure 2-8 Prefix tree generation pseudocode

The nodes of the prefix tree can also have values associated with them, although the only values that are commonly used are the values of the leaf nodes and some of the values of the inner nodes. In our case, we use the values to mark the end of a street name, which is usually a leaf node and more rarely an inner node, in the case when a street name is a prefix for another street name.

Dictionary search is one of the most common applications of the prefix tree. Searching a word in the prefix tree consists of traversing the prefix tree until a leaf node is reached, or until a node does not have any children whose key contains the desired letter. The recursive version of the prefix tree search algorithm is presented in ***Figure 2-9***.

```
FindString(root, string)
  IF (strlen(string) == 0) THEN
    RETURN root.index; //we have reached last node
  ELSE
    IF (string[0] is not the key of a child of root)
    THEN
      RETURN -1; //string is not found
    ELSE
      Set child as the child of root with the key string[0]
      RETURN FindString(child, substring(string,1))
```

Figure 2-9 Pseudocode of the recursive prefix tree search

In our implementation, we create a prefix tree for the street names of each municipality. Therefore, the street name detection becomes a dictionary search using a prefix tree. Because the Finnish street names usually end with a limited number of suffixes, the names were introduced in the prefix tree in reverse order and the search is done starting from the last letter. ***Figure 2-10*** gives an example of a pre-computed prefix tree.

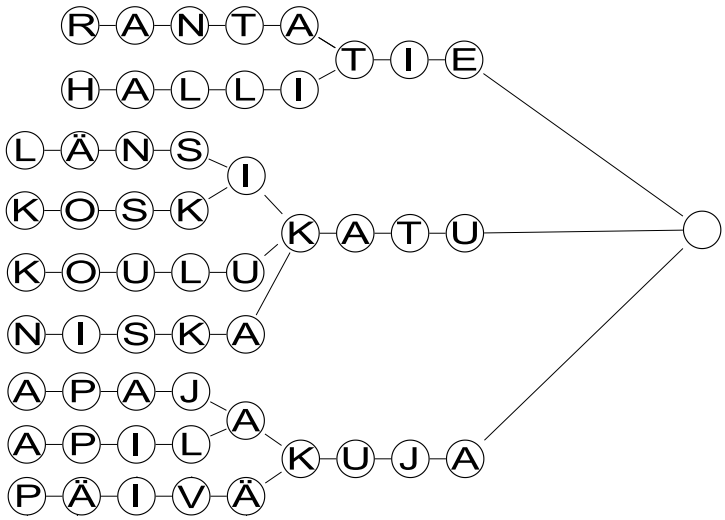

Figure 2-10 A sample prefix tree built from street names

Table 2-4 summarizes the prefix trees we computed for Finland and Singapore. Using prefix trees or other pre-built data structures to access street name data from the gazetteer decreases the needed storage space, in our case from 3 GB to 74 MB. We are still using the gazetteer for address validation and geo-coding.

Table 2-4 Prefix tree statistics

	Finland	**Singapore**
Maximum tree depth	34	14
Average tree depth	12.7	7.4
Average tree width	105	167
Average number of nodes per tree	2338	2335
Total size (MB)	74.4	0.18

2.6.4 Experiments

We tested the proposed method in the MOPSI location-based search engine using 20 different search locations and 10 keywords in English to construct <*keyword*, *city*> queries. We downloaded the content of the first 10 search results for each query of Google's search engine and the downloaded content was used as data input for the MOPSI prototype.

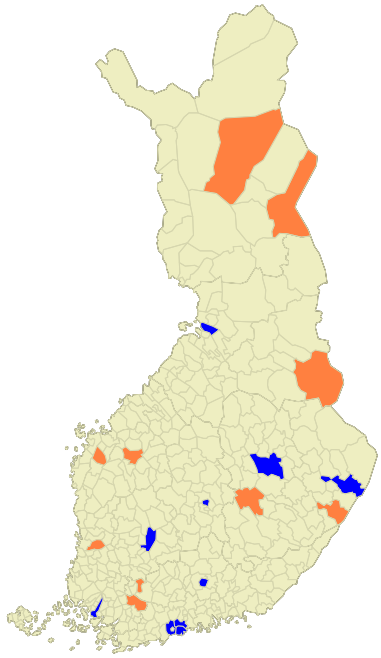

Figure 2-11 Locations used for experiments. The urban locations (blue): Espoo, Helsinki, Joensuu, Jyväskylä, Kuopio, Lahti, Oulu, Tampere, Turku, Vantaa; the rural locations (orange): Forssa, Kitee, Kuhmo, Laihia, Lapua, Pieksämäki, Salla, Sodankylä, Somero, Ulvila

The search locations were divided into two groups: 10 rural 10 urban municipalities (***Figure 2-11***), and the test keywords were divided into five commercial and five non-commercial ones (***Table 2-5***).

Table 2-5 Keywords used for experiments

Commercial	hotel, restaurant, pizzeria, cinema, car repair
Non-commercial	hospital, museum, police station, swimming hall, church

The addresses detected by each method were validated using our gazetteer database. The size of the downloaded data in the rural and urban municipalities is 13.9 and 11.2 MB, respectively.

Table 2-6 shows the average time for address detection and the number of detected addresses for the considered municipalities. The average time is calculated per query over all searches. According to the results, the proposed prefix tree method is considerably faster than our brute force method, and two to three times faster than our heuristic approach, which does not use the gazetteer. Typical search times of the prefix tree are less than one second per query.

Table 2-6 Average search times for the address detection

Method	Time (s)	Standard deviation	Number of validated addresses
	Rural municipalities		
Brute Force	3.01	2.43	3.7
Heuristic	1.54	1.15	2.5
Prefix Tree	0.51	0.35	3.7
	Urban Municipalities		
Brute Force	10.18	7.11	19.8
Heuristic	1.70	1.24	18.6
Prefix Tree	0.87	0.85	19.8
	Total		
Brute Force	6.59	6.40	11.8
Heuristic	1.62	1.20	10.5
Prefix Tree	0.69	0.68	11.8

The results also show that street density and city size do not affect the speed of the heuristic solution. In the prefix tree method, the average search time is slightly longer in urban municipalities (0.51 vs. 0.87 seconds). Street density affects the brute force method most because the search in street arrays is slower than in the prefix trees, resulting in more than three times slower search times in urban areas.

The detected addresses are validated using our gazetteer. The accuracy (number of validated addresses) is higher for the brute

force and prefix tree methods than for the heuristic method. This is because some of the street names do not follow our rules, having either unusual suffixes or multiple word names. The biggest difference between urban and rural municipalities is that the number and the density of streets are much larger in the urban municipalities and, therefore, the methods using gazetteer (prefix tree and brute force) are slower in rural municipalities. Nevertheless, the prefix tree method is the fastest even in this case.

In total, the proposed prefix tree method is on average twice as fast and 10% more accurate than the heuristic method. It reaches the same accuracy than the brute force search but uses only 10% of the processing time.

Our main goal in designing a gazetteer-based street address detector was to increase the accuracy in comparison to the fast heuristic method that was used in the earlier implementation [FKTS10]. This goal was achieved, as the proposed prefix tree solution is 57% faster and 10% more accurate, on average, than the heuristic solution. In comparison to brute force, it is 10 times faster. The resulting solution improves the speed and quality of web-page geo-referencing and removes one bottleneck for creating an efficient location-based search engine as the prototype *MOPSI search.*

2.7 EXTRACTING ASSOCIATED INFORMATION

We complement the search results by extracting a title and a representative image for each location we detect (see ***Figure 2-12***). We consider as title the name of the place or the service that is referenced by the address we detect. We consider the representative image to be the image that best describes the detected service. The location, along with its title, URL and representative image, forms a search result within our application.

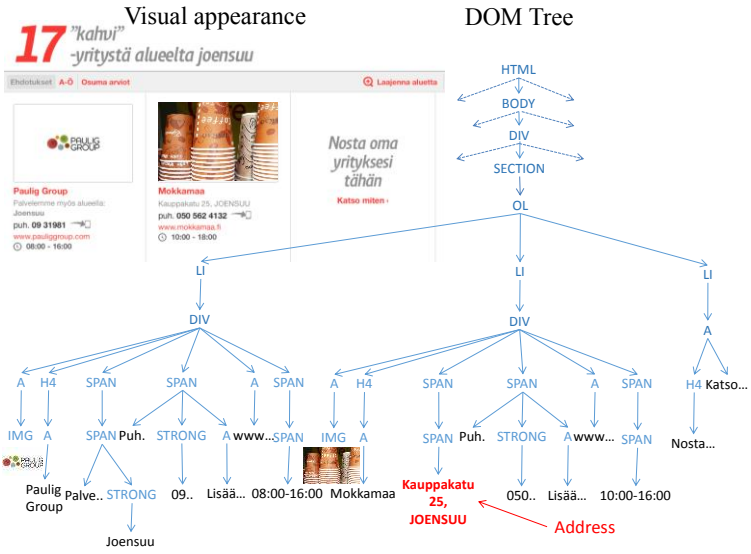

Figure 2-12 DOM tree of a website with location information. The text node that contains the location is marked in red

2.7.1 Extracting Representative Image

Images are used on web pages more than any other type of online content because they can transfer information to the user in a quick and efficient way. Although a large number of images are embedded into web pages, many of them are less relevant to the content of the web page, such as advertisements, navigational banners, icons and images that serve as section headings. A solution is needed to ignore the irrelevant images and find one representative image for the web page.

We define the representative image of a web page as the image that best represents the content of the page to the user. Representative images are important in many applications, especially in cases when bandwidth limitation restricts the total number of images that can be retrieved, or when building a visual category in which a single image must represent an entire category of documents and their associated content. This section presents the work that is detailed in [P5].

Figure 2-13 Image categories

We define five image categories based on the usage of the image within the web page (see ***Figure 2-13***) and rank them in the following priority order:

- *Representative*: images directly related to the content or the topic of the web page.
- *Logos*: recognizable images used to identify the company or institution that owns the website.
- *Banners*: images placed on a web page, either above, below or on the sides of the content. They are generally used for decoration. Headers and footers are classified in this category.
- *Advertisements*: images promoting products or services that are irrelevant to the topic or the content of the web page.
- *Formatting* and icons: images used to enhance the web page's visual appearance such as spacers, bullets, borders, backgrounds, or pictures used purely for decoration. We also include the small images which are not classified as logos and serve a functional purpose, such as icons which link to the home page or icons which are used for changing language.

src	http://www.ravintolakreeta.fi///images/banner.jpg
alt	--
title	--
from	css
format	jpg
width	945
height	202
size	190,890 px
aspect ratio	4.67
parent tag	
class	header

Figure 2-14 Image Features used

We retrieve the list of images from a website, by parsing its HTML, CSS and JavaScript source code. After retrieving the list of images, we analyze the following features of each image and its corresponding HTML element: URL, size, aspect ratio, HTML tag of the parent element, class and ID of the <img> tag and of the parent element (see ***Figure 2-14***).

Table 2-7 Rules used for image categorization

Category	Features	Keywords
Representative	Not in other category	
Logo	Parent is h1 or h2	logo
Banner	ratio > 1.8	banner, header, footer, button
Advertisements		free, adserver, now, buy, join, click, affiliate, adv, hits, counter
Formatting and icons	width < 100px height < 100px	background, bg, sprite

We are using the rules in ***Table 2-7***. In all categories, a predefined set of keywords is used. If any of these are found in the image URL or in the class name of the <img> tag and of the parent element, then the image is assigned to that category. In

addition to this, Banners and Formatting are also categorized according to image features such as image size and aspect ratio. Our rules were derived from direct observation, as, for example, banners usually have high aspect ratio, formatting and icon images are small, and logos, banners and advertisements have specific keywords in their class names.

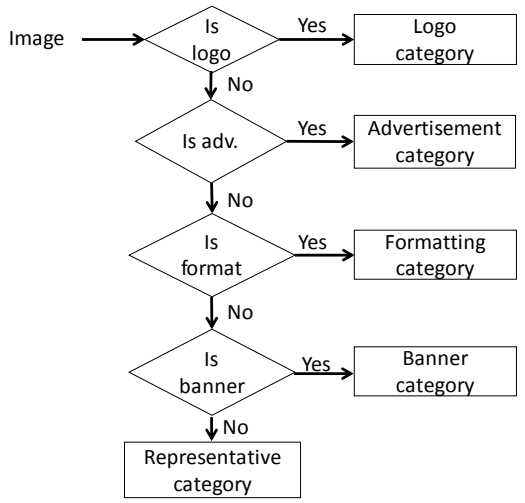

Figure 2-15 Decision tree used for image categorization

Note that the categories are overlapping, so that the same image may meet the requirements of multiple categories. However, in this case we are using a decision tree to assign the image category (see ***Figure 2-15***). An image can belong to the class of representatives only if it does not belong to any other category.

After image categorization, we rank the images within each category according to their size: the bigger the image size in pixels, the higher its rank. The categories are prioritized as shown in the beginning of this section. Neither HTML, CSS or JavaScript are prioritized, and the images in these file types are treated equally. We choose the representative image as the image with the highest rank.

2.7.2 Extracting Service Names

In order to extract the service names, we are using the DOM tree of the web page and we are splitting it into sub-trees until each

sub-tree contains just one postal address node. For each postal address node, we are scoring all the text nodes in their respective sub-trees. This section presents the work that is detailed in [GaTF15].

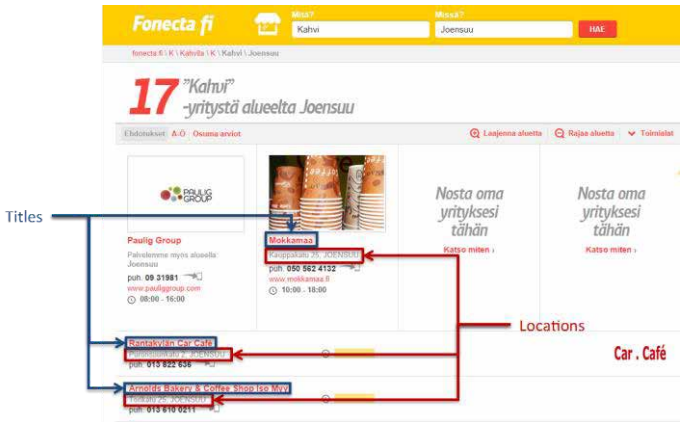

Figure 2-16 Web page that include several titles and addresses

In ***Figure 2-16***, there are several service titles located in the content of the web page that need to be detected, including "*Kauppakatu 25, Joensuu*". Usually service titles are accompanied by other information such as location, telephone numbers and opening hours. After the address has been detected, the service title is needed in order to generate a search result. We use the DOM tree of the web page and split it into sub-trees until each sub-tree contains only one postal address.

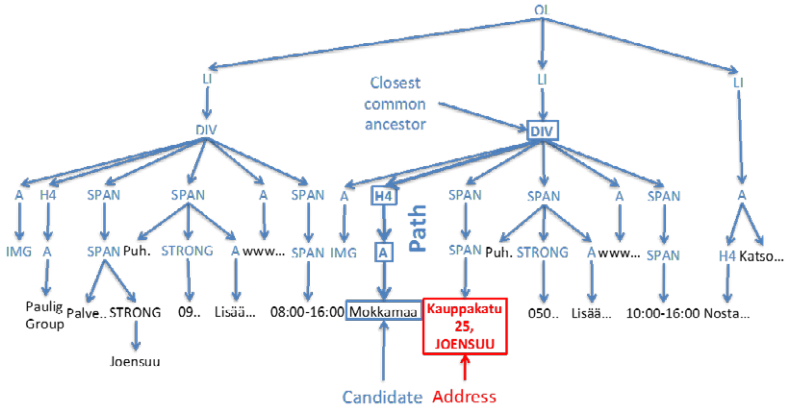

Figure 2-17 DOM sub-tree that contains the address of a service

An example of such a sub-tree is in **Figure 2-17**. For each sub-tree that contains one postal address, we score its text nodes based on visual appearance and distance to the postal address. Finally, we choose the node with the highest score (see **Figure 2-18** and ***Figure 2-19***).

Table 2-8 Scoring HTML tags

Tags	Score	Tags	Score
H1	+7	H5, H6, B, STRONG	+3
H2	+6	I, EM	+2
H3	+5	Others	0
H4, A	+4		

In order to score a text node, we first find its closest common ancestor to the postal address node. For example, in **Figure 2-17**, the detected address "*Kauppakatu 25, Joensuu*" and the candidate text node "*Mokkamaa*" have a DIV as the closest common ancestor. For calculating the visual appearance score, we first assign a score for each intermediary HTML tag along the path between the considered text node and the closest common ancestor node (in our example two SPAN tags) according to ***Table 2-8***.

Furthermore, we parse the CSS style sheet of the text node and we score the differences to the style sheet of the postal address using the following attributes: *color*, *background-color*, *font-size*, *font-weight*, *text-transform* (see ***Table 2-9***.). The perceptual color difference between the *color and background-color* attributes of a text node and the postal address node is scored from 0 to 10 by calculating the Delta E (CIE 2000) difference [LuCR01] and normalizing it to [0, 10].

Table 2-9 Scoring CSS attributes

CSS Attributes	Score
color, background-color	+ perceptual color difference (0 to 10)
font-size	+ (node font size - address node font size)
font-weight	+3 if bold or >500
text-transform	+5 if uppercase

The visual appearance score is calculated by summing the scores for HTML tags and the CSS style sheet differences:

$$S_A = S_{HTML} + S_{CSS} \quad (2.2)$$

Above, S_A is the appearance score, S_{HTML} is the score for an intermediary HTML tag, and S_{CSS} is the score for a CSS attribute difference.

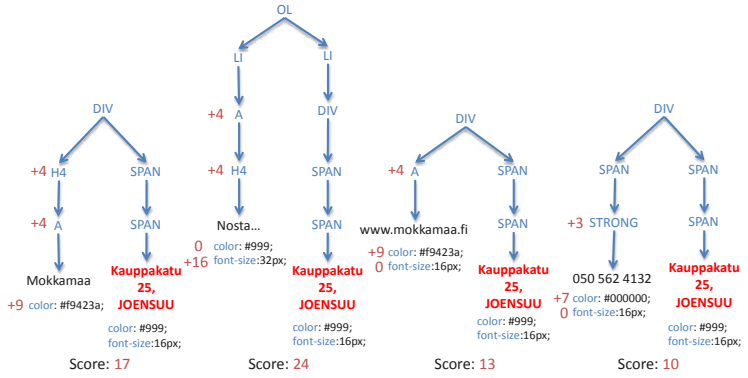

Figure 2-18 Scoring based on visual appearance

The scoring based on visual appearance for our example is detailed in **Figure 2-18** where we chose the four strongest candidates for the detected address, along with the subtree of the closest common ancestor. The candidate with the highest score, "*Nosta…*", is visually the most different from the address node, because the font size is significantly larger. It is also highlighted as a header and has a link. The second candidate, "*Mokkamaa*", is the actual title of the service. It is also highlighted as a header, has a link, and the text color is significantly different, but the font size is the same as that of the address node and its score is therefore lower.

We score the text nodes based on distance to the postal address by using a distance penalty. We use the DOM tree to determine the number of tags along the path from the address node to the closest common ancestor node (see ***Figure 2-19***). We divide the visual appearance score by this distance. We introduce this distance penalty because the nodes that are closer to the

postal address are more likely to contain information related to that address. Using the visual appearance score (2.2), the final score for a node becomes:

$$S_{NODE} = \frac{S_{HTML}+S_{CSS}}{d} \quad (2.3)$$

Above, d is the number of tags along the path from the postal address node to the closest common ancestor node.

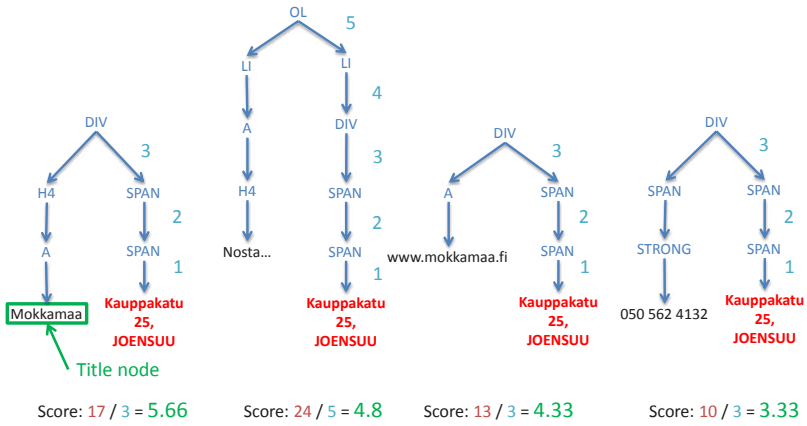

Figure 2-19 Adjusting scores according to distance penalty

Finally, we choose the text node with the highest score as the title of the service (see ***Figure 2-19***). In **Figure 2-18**, because of the distance penalty, the node with the highest visual score "Nosta…" has a smaller score that the node that is closer to the address "*Mokkamaa*".

2.8 EXPERIMENTS

In order to test our prototype application, we simulated the following scenario: the user is in the center of an urban or rural municipality and his search is restricted to that municipality. His targets can be commercial (i.e. restaurant) or non-commercial (i.e. police station).

These tests were performed in 2010 using the prototype described in [FTKH10]. We selected 10 Finnish urban municipalities and 10 Finnish rural municipalities (see ***Figure***

2-11). The urban municipalities were selected according to their size, and geographical location was taking into account when selecting the rural municipalities. The search was performed using 10 English test keywords which were divided into five commercial ones (hotel, restaurant, pizzeria, cinema, car repair) and five non-commercial ones (hospital, museum, police station, swimming hall, church), see ***Table 2-5***.

In addition, the same queries were submitted to two other location based web services: Google Maps and the Finnish Yellow Pages. According to [EgPa10], Google Maps uses multiple sources to provide the results. For instance, information submitted by local business owners or public directories, enhanced content such as reviews, photos submitted by users of various services, user generated content and other websites are crawled. Yellow Pages, on the other hand, is a public directory of local businesses and the data are maintained and verified by users and administrators.

The search results are sorted by distance and filtered according to each municipality. Duplicates were manually removed and the distance was calculated using the gazetteer. There is no standard methodology for testing the overall relevance of the location-based search results that considers the relevance of both topic and distance. We therefore formulated our own criterion based on the evaluation methodology proposed in [JaMo06]. First, we compare the total number of search results from the tested services without considering their relevance. Secondly, we compare the relevance of the search results by evaluating them as (1) *relevant*, (2) *somewhat relevant* or (3) *not relevant*.

As shown in ***Table 2-10***, our prototype provides more results than Google Maps or Yellow Pages, and is slightly more relevant (the smaller the better) than Google Maps, but less relevant than Yellow Pages. With the exception of urban non-commercial queries, our prototype gives more results. Unsurprisingly, Google Maps has more results than Yellow Pages for non-commercial queries and less for commercial ones.

Table 2-10 Number of results for our test scenario

Query type	MOPSI prototype	Google Maps	Yellow Pages
Rural non-commercial	69	29	0
Rural commercial	245	92	189
Urban non-commercial	148	413	37
Urban commercial	1412	813	1337
Total number of results	2352	1405	1597
Overall mean relevancy	1.59	1.66	1.28
Overall std. deviation	0.84	0.89	0.54
Overall std. error	0.02	0.02	0.01

Contrary to Google Maps and Yellow Pages, our prototype relies on the relevance of the results of the external search engine. The search results are basically the addresses found in the results of the external search engine, whilst Google Maps has multiple criteria for relevance and Yellow Pages is controlled by human evaluators. We therefore compare the relevance of the search engines on an average basis, using a combination of two keyword categories for each comparison. A comparison of rural municipals and non-commercial keywords resulted in our search engine (MOPSI) having a mean value less than that of Google Maps (2.35 comparing to 2.48), whilst Yellow Pages did not return any results. This indicates a relatively high number of relevant results obtained by our search engine. The same method was used in **Table 2-11**, in which the results show a diverse number of relevant links from the MOPSI search.

Table 2-11 Mean relevance for our test scenario

Query type	MOPSI prototype	Google Maps	Yellow Pages
Rural non-commercial	2.35	2.48	0
Rural commercial	1.71	1.33	1.36
Urban non-commercial	2.20	2.17	1.59
Urban commercial	1.46	1.41	1.27
Overall mean relevancy	1.59	1.66	1.28

The results show that the relevance of the MOPSI search engine is close to Google Maps, except for the rural municipalities with non-commercial keywords, for which we get a higher number of results, but their overall relevance is smaller. Yellow Pages is the most relevant service, but has the lowest number of results (except for commercial urban keywords). In urban areas, the number of results by the MOPSI search engine is close to Google Maps and Yellow Pages, whilst in rural areas it is higher.

2.8.1 Observations and known problems

One of the main problems is that the search can produce a vast number of irrelevant results. Mobile devices have restricted resources (data transfer bandwidth, small display) and often poor usability. The application should therefore be able to filter out flawed or less relevant data much better than a web-based application to make the browsing of the search results easier. Because of this, the current version downloads only a limited amount of links but further ideas to improve this would be desirable.

Another problem is that, unlike normal web searches, we allow the results to include parts of web pages. This is very useful for finding services that do not have their own web page but exist in ad-hoc service directories such as www.pizza-online.fi. However, an open problem is how to extract only the relevant part from a web page without any prior knowledge about its structure.

Despite the previously mentioned problems, many positive results are also achieved. When it comes to non-commercial services, web pages are good repositories of location relevant information. In the rural areas of Finland where businesses are small, more services are found from web pages than from commercial databases.

2.8.2 Conclusions

The concept of a location-based search engine was outlined, and its design issues and problems were discussed. The MOPSI prototype implementation in Finland was demonstrated with qualitative comparison using typical search examples. The idea itself was also generalized worldwide, although the practical implementation has some issues such as the accuracy of the gazetteer and the detection of the address elements. The chosen address detection method (especially the street name detection), relies on the accuracy of the OpenStreetMap gazetteer. It needs to find the exact string match, and can support multi-word street names or addresses as the street names are indexed as strings that can also contain spaces, but it does not support any variation on the order of the words in a street name, mistyped words, or abbreviations. Methods such as term normalization [AhBo08a] should be considered in future work to improve address detection, but they also require changes in querying the OpenStreetMap gazetteer.

The results indicate that the proposed approach has much potential for practical applications. Most of the problems are related to technical matters and implementation issues. For instance, we use real-time search and page parsing, whereas commercial solutions such as Google can use large computer capacities and avoid computational problems by pre-processing, huge storage (cache), and indexing.

3 Location-Based Mobile Orienteering Game

Combining gaming and physical activity has always been a desirable goal. Furthermore, using mobile games instead of "real world" applications is better for training people in using new technologies. This is because mobile games provide a virtual environment in which users are encouraged to experiment.

One the one hand, using use of mobile devices for playing games is a very popular and long-established idea. On the other hand, mobile gaming that uses the player's location and involves physical activity is a recent idea, although it is starting to gain popularity. With the fast development of computing devices, mobile location-based gaming has evolved from using wearable computers and head-mounted displays [PiTh02] to using mobile phones with access to the Internet and GPS positioning.

We introduce a mobile location-aware game, O-Mopsi, which is based on the classical concept of orienteering and exploits the data available in a geo-tagged user generated photo collection, available at http://cs.uef.fi/mopsi/photos. Users can create games for others to play by defining a set of targets. The players need to visit all the targets in order to complete the game. O-Mopsi was first introduced in [P6]; a more detailed description of the game and its components is documented in [Wan14].

The game can be played using a mobile application that is available for all the modern mobile operating systems (Symbian, Windows Phone, Android and iOS). Players are free to choose how to play the game, regardless whether they are walking and running or using different transportation modes such as a bicycle.

O-Mopsi is available at http://cs.uef.fi/o-mopsi. The mobile client has functionalities such as plotting the targets' photos on the map, displaying compass data and modifying sound

frequency and pitch to indicate the distance to the selected target. The web interface allows game management, real-time player tracking, post-game trail analysis and suggests reference routing by using either greedy heuristics or an ant colony-based optimization.

3.1 RELATED WORK

One of the first examples of location-aware games is *Pirates!* [BFHL01], which shows an example how the player's physical location is integrated into the computer game's design and is used to trigger game events. A large number of location-aware games are adaptations of traditional board games or computer games, including *Quake* [PiTh02], *Pacman* [CGLF04], *Tic-Tac-Toe* [ScKM05], *Monopoly* [LiOO08] *Chase and Catch* [MHKT09] and *Snake* [ChSi12]. The extension of video games from the traditional virtual world to real locations using mobile devices is also discussed in [RMCE06]. Other approaches combine mobile gaming with education and game reality with physical reality.

Savannah, [FJSR04], is a mobile game based on animal behavior and aimed at children, who use GPS enabled devices to navigate in a virtual savannah. The children can observe animals and must learn how to survive. Education is also the goal of *Skattjakt* (*Treasure Hunt*), [SpMi08], which shows that mobile outdoor games are well suited for novel learning activities that involve physical motion, problem solving, inquiry and collaboration. *Treasure Hunt* encourages players to get physically active by solving a mystery surrounding a castle located on the university's campus. Another approach to education and treasure hunting is the game *Tidy City* [WeBO12], which requires solving location-based riddles and city exploration.

"*Song of the North*" [LHNL04] is a game inspired by Nordic mythology that combines game reality and physical reality. The physical world is combined with a virtual spirit world, in which the player can interact by using a mobile device. Another example of mixed reality is "*Can you see me now?"* [BCFD06], in which players are chased by runners through a virtual model of

a city. The runners are professional performers equipped with Wi-Fi and GPS. The rules are similar with the game of catch, with the difference that the "runners" have to traverse actual city streets in order to capture the online (virtual) players.

In a location-aware game, photos can be used in various ways, such as a means of interaction [SuKo06], to provide additional information about game targets, such as O-Mopsi and in other games such as *See It* [NeJu12] or to produce useful information for other application, such as *CityExplorer* [MMSK08].

CityExplorer applies the "games with a purpose" paradigm to users with mobile devices that have to take geo-referenced photos, identify points of interest and categorize them semantically. O-Mopsi uses the location-based data that is produced by other applications, but also encourages users to create games with a purpose by offering the possibility to create sightseeing tours.

O-Mopsi can also be viewed as an exergame, which are video games that require physical activity to play. According to [BoYa13], there are several categories of exergames: mobile location-based games, mobile location-aware fitness and sports applications with social and games features, and location-aware sports gadgets. O-Mopsi belongs to the first category.

3.2 GAME RULES

The goal of the game is to visit a set of targets in the shortest time. A target is identified by the following attributes: location, photo and a short description. A game can be created by selecting photos from the MOPSI photo collection using our web interface.

A game starts when the player visits the first target. The order of targets is not fixed and the player can freely choose which targets to visit. The game ends when the player has visited all the targets. To visit a target, a player has to be closer than 20 meters from the target's location. This threshold was chosen taking into consideration GPS inaccuracies that can occur either when collecting geo-tagged photos or when playing a game.

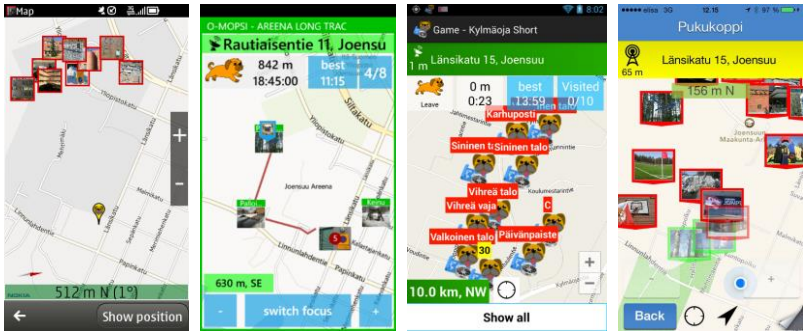

Figure 3-1 Game screen in the 4 supported mobile platforms: Symbian, Windows Phone, Android and iOS

Game results and players' progress can be viewed online in real time using the web interface, which also includes tools for game analysis such as calculating the shortest path. Players are ranked in the order of the completion time. The total distance, the starting target or the order of visiting targets do not affect ranking.

The game shares similarity with the concepts of orienteering and geocaching [Came04], which both require identifying and visiting a number of targets.

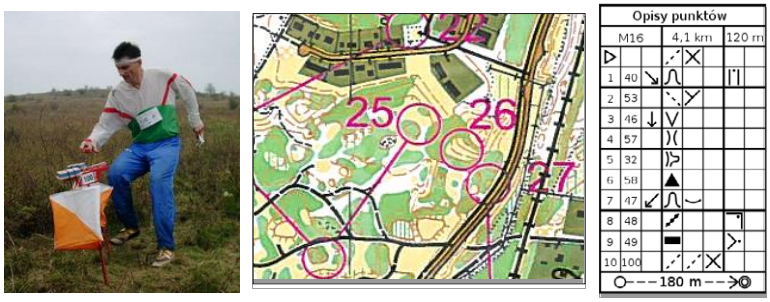

Figure 3-2 Orienteering: control point (left), map (middle) and control description sheet

Orienteering requires navigating from point to point in a predefined order using a map and a compass. Unlike orienteering, in O-Mopsi the targets can be visited in any order, encouraging players to develop different strategies and to choose different starting points. Furthermore, in order to make gameplay easier, O-Mopsi uses digital maps and GPS so that the player can

visualize his position on the map while playing, features which are not available in orienteering.

Figure 3-3 A classic geocache box (source: https://en.wikipedia.org/wiki/Geocaching#/media/File:Classic_Geocache.jpg)

Geocaching requires finding a "hidden treasure" (a collection of things placed in a container called geocache and placed in a location available to the public). The GPS location of the cache is published online and other players need to find the cache and replace an item from the collection with another. O-Mopsi also requires finding a certain GPS location defined by another player, but the location is additionally identified by a photo. Furthermore, in Geocaching, the time does not matter and targets are not grouped into games or other entities.

O-Mopsi uses virtual targets, whereas orienteering and geocaching targets are physical objects.

3.3 WEB INTERFACE

The O-Mopsi web interface can be used for creating and managing a game, displaying the proposed shortest path of a game, displaying game results and viewing real-time player progress.

The architecture of the website, along with the main functions, is presented in ***Figure 3-4***. For displaying games and targets we

use Google Maps, the web interface is developed using PHP and JavaScript. We store game data in a MySQL database with spatial extensions.

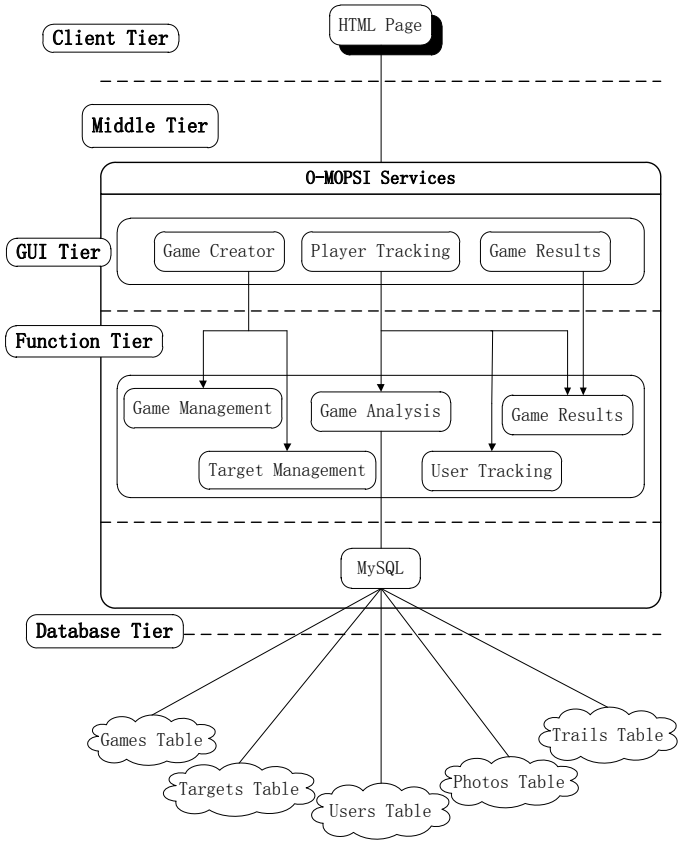

Figure 3-4 Architecture of the O-Mopsi website

The client tier contains the HTML page that is presented to the user. The middle tier is composed of the Graphical User Interface (GUI) tier and the function tier. The GUI tier includes interfaces for the creation of the game, for the live tracking of the players and for the game results. The function tier includes all the functions needed for the server logic: game and target management, game analysis, user tracking and game results. Finally, the database tier consists of the tables needed to store game data (games, targets, users, photos and trails).

Using the web interface, player can manage games with operations such as creating, editing and deleting. An example of such game is shown in ***Figure 3-5***.

Figure 3-5 Sightseeing tour game in Singapore

A typical workflow of creating and editing a game is documented in ***Figure 3-6***, where a user can create a new game and edit or delete games.

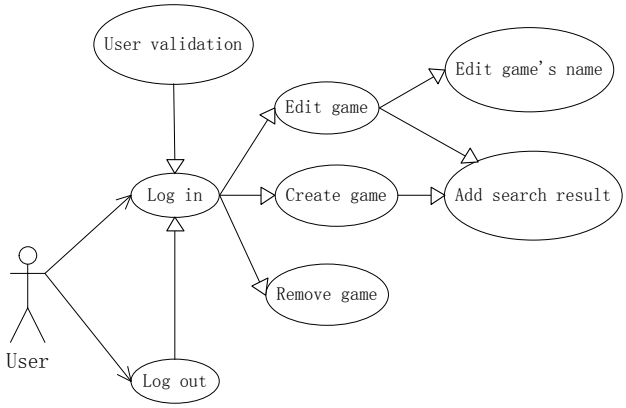

Figure 3-6 Typical workflow for game management

For a game to be playable, the quality of targets is important. Therefore, we created a set of guidelines for creators to follow:

- Outdoor targets only: Targets can be small, such as a postbox, logo on a door or small detail on a building or big, such as a shop, a bridge or another easy to find place. Selecting people, animals and indoor objects as targets is not recommended.
- Reachable and clear target: Players must have a certain object or location to look for. A photo of a large building that is taken from distance could not clearly indicate the place that needs to be visited by the player. Smaller items such as entrances are easier to identify. Also, targets in places that are temporarily open to the public, such as stadiums, are discouraged.
- Accuracy of location: In our photo collection, there are photos which are taken far away from the object and photos that have inaccurate locations caused by GPS errors. This is why we offer the player the option to fine-tune the location on the map.
- Permanent targets: Some targets, such as vegetation, piles of snow, advertisement signs are temporary or season-dependent and we do not recommend choosing them.
- Distance between the targets: If targets are too far from each other, the player will lose interest in finding them. According to the feedback we got from SciFest, most of the players give comments that finding targets is the most interesting thing during the game, but there were some cases when players gave up the game because one of the targets was too far away from the other ones.
- Number of targets: A game should have at least three targets. The game creator can choose as many targets as he or she wishes and can use our analysis tool to check the reference route length. However, choosing a large number of targets could discourage inexperienced players.
- Short but descriptive names for targets: The map view of the mobile screen cannot accommodate long names. We recommend short and descriptive names for targets that should give a hint to the players to locate them. Names "Target 1", "No.1" and empty names must be avoided.

- Descriptive name for the game: A suitable game name can attract more players. For example, game names can be: *Postbox maniac*, *Joensuu Cafes* or *Niinivaara sightseeing*, which give a clear description of the purpose of the game.
- Language: Currently we do not support localization, therefore English is preferred, but the local language can also be used.

3.4 GAME CLIENT

The game client is available for most modern mobile operating systems. After registering or logging in, the player can choose to join a new game or continue an existing active game.

During gameplay, the player's location is tracked and stored on the server. The main screen of the application (see ***Figure 3-7***) shows the current location, accuracy and statistics such as playing time, distance and speed. It also contains shortcuts to the map, an option to highlight the closest target, to view the full list of targets and the game results (see ***Figure 3-8***).

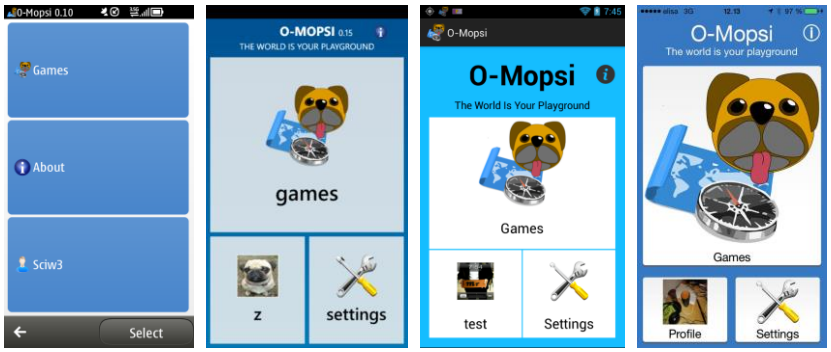

Figure 3-7 The start screen of the application

A target can be identified by its photo, which can be highlighted on the map (see ***Figure 3-9***). To aid navigation, the application displays the distance and bearing to the selected target along with player's orientation taken from the phone's compass sensor.

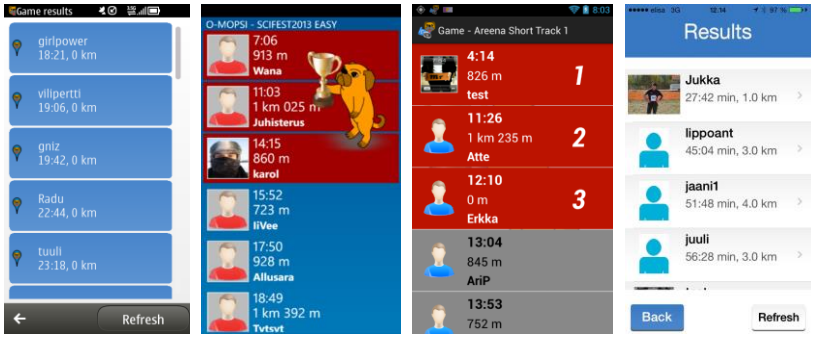

Figure 3-8 Game results screen

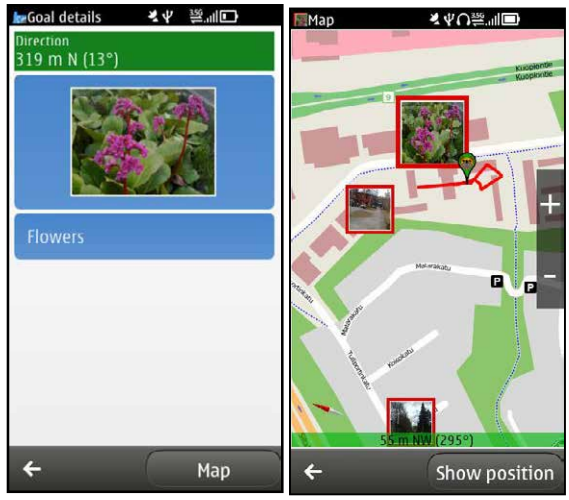

Figure 3-9 Viewing target details (left) and highlighting the chosen target on the map, along with the player's trail (right).

Additionally, the player is guided by sounds while the map is open. When the player approaches a target and is closer than 500m, a beeping sound is played at fixed intervals. The interval between sounds is inversely proportional to the distance, starting from 5 seconds for 500 meters and decreasing by 1 second every 100m (see ***Table 3-1***). The sound frequency also increases or decreases as the player becomes closer or further away from the target. Visiting the target turns off the sound guidance. We implemented this option in order to address an important issue in location-based games: the player's safety. We tried to minimize

the time the player has to look at the phone screen while he is heading for the selected target.

Table 3-1 The sound interval for different distance levels

Distance	Sound interval	Distance	Sound interval
> 500 m	no sound	100-200 m	2 s
400-500 m	5 s	60-100 m	1 s
300-400 m	4 s	20-60 m	0.5 s
200-300 m	3 s	< 20 m	"found" sound

3.5 FEEDBACK

O-Mopsi is mainly designed for the SciFest festival (www.scifest.fi), which is an annual international festival in which thousands of school kids, high school students, and teachers discover new experiences and learn about science, technology and the environment [JoKo10]. SciFest is organized in the city of Joensuu, Finland in the month of April.

O-Mopsi was presented during the 2011–2014 editions of the festival. Because of the limited availability of mobile phones with GPS and data connection, the players were organized into teams. After the game, the playing teams completed a short feedback survey.

Table 3-2 Players' ratings of the game

Feedback	Very good	Good	Needs improvement	Bad
Scifest 2011	3	6	0	0
Scifest 2012	3	7	0	2
Scifest 2013	2	21	6	0
Scifest 2014	8	19	3	0
Scifest 2015	9	9	1	0
Total	25	62	10	2

Feedback (see ***Table 3-2***) shows that the players mostly rated the game as being good or very good. According to the users, game rules are easy to understand and playing the game is enjoyable. Negative feedback was caused by software problems, or in some cases by problems in finding a GPS signal.

3.6 CONCLUSIONS

We presented a mobile location-aware game that is based on the classical concept of orienteering and the data available in a geo-tagged user-generated photo collection. The game can be played using a mobile application available for most of the current mobile operating systems, offering functionalities such as plotting the target photos on the map, displaying compass data, and modifying sound frequency and pitch to indicate the distance to the selected target. The web interface allows game management, real-time player tracking, post-game trail analysis and suggests game solutions. Testing O-Mopsi in a real-world situation during an international festival produced positive feedback.

4 Summary of the Contributions

All our proposed methods have been implemented in the MOPSI framework, which uses real data collected by means of our mobile applications.

In **[P1]** we define four aspects of relevance in sharing location-based media: location, time, content and social network. We study how these aspects of relevance appear in three media sharing platforms: Picasa, Flickr and MOPSI. Experiments show that the location is an important aspect, but not the only one. These results have been used for developing a location-based search and recommendation system.

In **[P2]** we propose a location-aware search engine for the mobile environment. We use a meta-search approach: we retrieve the results of a search engine that is not location-aware and post-process them by detecting locations and associated information. This paper sketches the overall scheme of this MOPSI search engine prototype for Finland, and defines all the core elements needed to make it happen. For rural commercial websites it provides better results than Google Maps and Yellow Pages. Since then, the MOPSI application has been extended in four different platforms that include the search engine component: Nokia Symbian, Windows phone, Android and iOS.

In **[P3]**, we describe the address detection used in the MOPSI engine. We find locations by detecting postal addresses using prefix trees and a gazetteer. Address detections start with street name detections, which are aggregated with other address elements in order to build address candidates that are validated using our gazetteer. We study both heuristic and gazetteer-based methods to identify street names, showing that a gazetteer-based prefix tree search method is the fastest and most accurate. This

method follows the general workflow of most of the state-of-the-art address detection methods, but it is different because it uses portions of the DOM tree and does not impose a certain order of elements, which helps generalization.

[P4] improves the methods in **[P2]** and **[P3]** by replacing plain text extraction with the processing of the DOM representation, and by improving the methods for extracting the title and representative image for each search result. We integrate processes of geo-referencing, geocoding and geo-tagging into a unified location-based system that is able to provide relevant and close information the user. The implementation is not limited to a specific geographic location, although it works for addresses that follow a structure similar to Western European addresses and the accuracy is influenced by the quality of the gazetteer and the string matching method we use.

In **[P5]** we study how to select a representative image to a web page. This solution is needed as part of the summarization of the web page found by the MOPSI search. Despite it being n use, also on social media sites, no good solutions were found in literature. The closest examples can be found in Facebook and Google+, which use simple heuristic solutions for selecting the image with the web link that the user wants to share. We propose a rule-based method that provides 64% accuracy, which is better than that of Google+ (48%) and Facebook (39%). Besides summarizing the MOPSI search results, the proposed method is directly applicable to those two social media sites as well.

In **[P6]**, we propose a location-aware mobile game called O-MOPSI that promotes physical exercise by applying concepts from the classical game of orienteering and uses geo-tagged photo collections created by users. In order to complete a game, a player must visit a set of targets, with photos chosen from a user-generated geo-tagged database. The main contributions are that the game is created by using data from a user-generated multimedia collection and that the players can freely chose the order of the targets, so we encourage them to try different strategies to solve the shortest and fastest path problem. Our

game can be also used for creating sightseeing tours in the form of an orienteering game. O-MOPSI has been presented at an annual international festival which is aimed at introducing science and technology to school children, and has received positive feedback from the players.

5 Conclusions

We have studied how location-based data can be used in two types of applications: web data mining and mobile games. Location is an important factor in personalizing applications where user context is needed.

Typically, websites do not include explicit location data, but postal addresses are the most common way locations are included into web pages. We have proposed a method to detect location-based data, starting from identifying addresses and using them to detect additional information such as title and representative image. This approach works well especially with commercial services and service directories, because they have address information written in a clear way, but it depends on the quality of the gazetteer we use. Detecting service title, address and representative image allows us to provide meaningful results in a compact and informative way that is suitable for the mobile users.

In future work, we will consider how to detect what type of website contains location data. It can be a single entity (person, business or place), a chain of services (such as restaurants, banks or shops) or a service directory that displays all the items that are related to a certain field or area (such as restaurants in one city or local businesses in an area). This would improve the quality of the search results, especially on the websites that are not service directories, because each type of website embeds location data in a different way and because in single service websites the name of the service is often not close to the address. Furthermore, our address detection method needs to be improved to support abbreviations and mistyped words. This can be done by improving our string matching algorithm that currently supports only exact matches. Further study needs to be done for other regions than Finland, which might have higher variation in how addresses are written.

We adapted the classical game of orienteering to be played on mobile phone using GPS, digital maps and geo-tagged photos our users have collected. Our goals were to provide a fun way for doing physical exercise, to create an environment that helps learning map navigation and finding the shortest path, and to encourage our users to collect geo-tagged photos and to create games. We have presented the game at a science festival for five years in a row and we plan to improve it by conducting larger-scale testing and using the feedback data to improve its usability aspects. Aside from the technical and conceptual issues from creating the game, more research is needed to measure the impact in improving the players' attitude towards physical exercise and towards the adoption of new technologies.

Bibliography

[Abit97] Abiteboul, S.: "Querying semi-structured data". In: *Database Theory—ICDT'97*, *Lecture Notes in Computer Science*. vol. 1186. Berlin, Heidelberg : Springer Berlin Heidelberg, 1997, pp. 1–18

[AhBo08a] Ahlers, D. and S. Boll: "Retrieving address-based locations from the web". In: *Proceeding of the 2nd international workshop on Geographic information retrieval - GIR '08*. New York, New York, USA : ACM Press, 2008, p. 27

[AhBo08b] Ahlers, D. and S. Boll: "Urban web crawling". In: *Proceedings of the first international workshop on Location and the web - LOCWEB '08*. New York, New York, USA : ACM Press, 2008, pp. 25–32

[AhCo75] Aho, A. V. and M. J. Corasick: Efficient string matching: an aid to bibliographic search. In: *Communications of the ACM* vol. 18 (1975), pp. 333–340

[AHSS04] Amitay, E., N. Har'El, R. Sivan, and A. Soffer: "Web-a-where". In: *Proceedings of the 27th annual international conference on Research and development in information retrieval - SIGIR '04*. New York, New York, USA : ACM Press, 2004, p. 273

[BCFD06] Benford, S., A. Crabtree, M. Flintham, A. Drozd, R. Anastasi, M. Paxton, N. Tandavanitj, M. Adams, et al.: Can you see me now? In: *ACM Transactions on Computer-Human Interaction* vol. 13 (2006), Nr. 1

[BRWY11] Bennett, P. N., F. Radlinski, R. W. White, and E. Yilmaz: "Inferring and using location metadata to personalize web search". In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011, pp. 135–144

[BFHL01] Bjork, S., J. Falk, R. Hansson, and P. Ljungstrand: "Pirates! using the physical world as a game board". In: *13th International Conference on Human-Comupter Interaction*. Tokyo, Japan, 2001, p. 8

[BLMD07] Borges, K. A. V., A. H. F. Laender, C. B. Medeiros, and C. Davis Jr.: "Discovering geographic locations in web pages using urban addresses". In: *GIR '07 Proceedings of the 4th ACM workshop on Geographical information retrieval*. Lisbon, Portugal, 2007, pp. 31–36

[BoYa13] Boulos, M. N. K. and S. P. Yang: Exergames for health and fitness: the roles of GPS and geosocial apps. In: *International journal of health geographics* vol. 12 (2013), Nr. 1, p. 18

[BCGG99] Buyukkokten, O., J. Cho, H. Garcia-Molina, L. Gravano, and N. Shivakumar: "Exploiting Geographical Location Information of Web Pages". In: *In Proceedings of the ACM SIGMOD Workshop on the Web and Databases (WebDB'99)*. Philadelphia, Pennsylvania, USA, 1999, pp. 1–6

[CYWM03] Cai, D., S. Yu, J.-R. R. Wen, and W.-Y. Y. Ma: VIPS: A vision-based page segmentation algorithm. In: *Microsoft technical report, MSR-TR-2003-79* (2003), pp. 1–29

[CaWJ05] Cai, W., S. Wang, and Q. Jiang: Address extraction: Extraction of location-based information from the web. In: *Web Technologies Research and Development* (2005), pp. 925–937

[Came04] Cameron, L. S.: *The geocaching handbook* : Globe Pequot, 2004

[CQXW05] Can, L., Z. Qian, M. Xiaofeng, and L. Wenyin: "Postal address detection fromweb documents". In: *Proceedings of International Workshop on Challenges in Web Information Retrieval and Integration. WIRI'05.*. Tokyo, Japan, 2005, pp. 40–45

[ChLi10] Chang, C. H. and S.-Y. Y. Li: "MapMarker: Extraction of postal addresses and associated information for general web pages". In: *International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*. vol. 1. Toronto, Canada, 2010, pp. 105–111

[ChZF10] Chen, J., Q. Zhao, and P. Franti: "Smart Swap for more efficient clustering". In: *The 2010 International Conference on Green Circuits and Systems*. Shanghai, China, 2010, pp. 446–450

[ChXF12] Chen, M., M. Xu, and P. Fränti: A fast O(N) multiresolution polygonal approximation algorithm for GPS trajectory simplification. In: *IEEE Transactions on Image Processing* vol. 21 (2012), pp. 2770–2785

[CGLF04] Cheok, A. D., K. H. Goh, W. Liu, F. Farbiz, S. W. Fong, S. L. Teo, Y. Li, and X. Yang: Human Pacman: A mobile, wide-area entertainment system based on physical, social, and ubiquitous computing. In: *Personal and Ubiquitous Computing* vol. 8 (2004), pp. 71–81

[ChSi12] Chittaro, L. and R. Sioni: "Turning the classic snake mobile game into a location-based exergame that encourages walking". In: *Lecture Notes in Computer Science*. vol. 7284, 2012, pp. 43–54

[CSLJ09] De Choudhury, M., H. Sundaram, Y. R. Lin, A. John, and D. D. Seligmann: "Connecting content to community in social media via image content, user tags and user communication". In: *Proceedings of 2009 IEEE International Conference on Multimedia and Expo, ICME*. New York, NY, USA, 2009, pp. 1238–1241

[Delo10] Delort, J.-Y.: "Vizualizing Large Spatial Datasets in Interactive Maps". In: *2010 Second International Conference on Advanced Geographic Information Systems, Applications, and Services*. St. Maarten, Netherlands Antilles : IEEE, 2010, pp. 33–38

[DiGS00] Ding, J., L. Gravano, and N. Shivakumar: "Computing geographical scopes of web resources". In: *Proceedings of the*

26th International Conference on Very Large Data Bases. Cairo, Egypt, 2000, pp. 545–556

[DoHu12] Dou, W. and J. Hu: "Automated Web Data Mining Using Semantic Analysis". In: *Advanced Data Mining and Applications*, 2012, pp. 539–551

[EgPa10] Egnor, D. and E. Pasztor: Generating structured information, Google Patents (2010). — US Patent 7,788,293

[FJSR04] Facer, K., R. Joiner, D. Stanton, J. Reid, R. Hull, and D. Kirk: Savannah: Mobile gaming and learning? In: *Journal of Computer Assisted Learning* vol. 20 (2004), pp. 399–409

[FLMN10] Florczyk, A. J., F. J. López-Pellicer, P. Muro-Medrano, J. Nogueras-Iso, and F. J. Zarazaga-Soria: Semantic selection of georeferencing services for urban management. In: *Journal of Information Technology in Construction* vol. 15 (2010), pp. 111–121

[FKTS10] Fränti, P., J. Kuittinen, A. Tabarcea, and L. Sakala: "MOPSI location-based search engine". In: *Proceedings of the 2010 ACM Symposium on Applied Computing - SAC '10*. Sierre, Switzerland, 2010, p. 872

[FTKH10] Fränti, P., A. Tabarcea, J. Kuittinen, and V. Hautamäki: "Location-based search engine for multimedia phones". In: *2010 IEEE International Conference on Multimedia and Expo, ICME 2010*. Singapore, 2010, pp. 558–563

[GaTF15] Gali, N., A. Tabarcea, and P. Franti: Rule-based technique for extracting representative title for services detected in webpage. In: *manuscript* (2015)

[GoWK07] Goldberg, D. W., J. P. Wilson, and C. A. Knoblock: From text to geographic coordinates: the current state of geocoding. In: *URISA journal* vol. 19 (2007), Nr. 1, pp. 33–46

[HaFM02] Hariharan, G., P. Fränti, and S. Mehta: "Data mining for personal navigation". In: *AeroSense 2002*. Orlando, Fl, USA, 2002, pp. 355–365

[HeMS13] Hess, B., F. Magagna, and J. Sutanto: Toward location-aware Web: extraction method, applications and evaluation. In: *Personal and Ubiquitous Computing* vol. 18, Springer (2013), Nr. 5, pp. 1047–1060

[HiFZ99] Hill, L. L., J. Frew, and Q. Zheng: The Implementation of a Gazetteer in a Georeferenced Digital Library. In: *D-Lib Magazine* vol. 5 (1999), pp. 1–17

[HuLR05] Hu, Y.-H., S. Lim, and C. Rizos: *"Georeferencing of Web Pages Based on Context-Aware Conceptual Relationship Analysis"*. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.434.1333&rep=rep1&type=pdf

[HuFi10] Huitema, P. and P. Fizzano: "A crawler for local search". In: *4th International Conference on Digital Society, ICDS 2010*. St. Maarten, Netherlands Antilles, 2010, pp. 86–91

[JaMo06] Jansen, B. J. and P. R. Molina: The effectiveness of Web search engines for retrieving relevant ecommerce links. In: *Information Processing & Management* vol. 42 (2006), Nr. 4, pp. 1075–1098

[JoKo10] Jormanainen, I. and P. Korhonen: "Science festivals on computer science recruitment". In: *Proceedings of the 10th Koli Calling International Conference on Computing Education Research*. Koli, Finland, 2010, pp. 72–73

[KaKa09] Kasemsuppakorn, P. and H. A. Karimi: "Pedestrian network data collection through location-based social networks". In: *5th International Conference on Collaborative Computing: Networking, Applications and Worksharing*. Washington D.C., USA, 2009

[Küpp05] Küpper, A.: *Location-Based Services: Fundamentals and Operation*. 1st. ed. : Wiley Online Library, 2005

[LHNL04] Lankoski, P., S. Heliö, J. Nummela, J. Lahti, F. Mäyrä, and L. Ermi: "A Case Study in Pervasive Game Design: The Songs of North". In: *Proceedings of the third Nordic conference on Human-computer interaction - NordiCHI '04*. Tampere, Finland, 2004, pp. 413–416

[LeLM07] Lee, H. C., H. Liu, and R. J. Miller: "Geographically-Sensitive Link Analysis". In: *IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)*. Sillicon Valley, USA : IEEE, 2007, pp. 628–634

[LeLL13] Leung, K. W. T., D. L. Lee, and W.-C. Lee: Pmse: A personalized mobile search engine. In: *IEEE Transactions on Knowledge and Data Engineering* vol. 25, IEEE (2013), Nr. 4, pp. 820–834

[LiOO08] Li, M., M. O'Grady, and M. P. O'Hare, Gregory: "Geo-gaming: The mobile monopoly experience". In: *Fourth International Conference on Web Information Systems (WEBIST2008),*. Madeira, Portugal, 2008, pp. 220–223

[LGCZ12] Li, W., M. F. Goodchild, R. L. Church, and B. Zhou: "Geospatial Data Mining on the Web: Discovering Locations of Emergency Service Facilities". In: *Advanced Data Mining and Applications* : Springer, 2012, pp. 552–563

[LiRG10] Liu, C., P.-L. P. Rau, and F. Gao: Mobile information search for location-based information. In: *Computers in industry* vol. 61, Elsevier (2010), Nr. 4, pp. 364–371

[LiMM10] Liu, W., X. Meng, and W. Meng: ViDE: A Vision-Based Approach for Deep Web Data Extraction. In: *IEEE Transactions on Knowledge and Data Engineering* vol. 22, IEEE (2010), Nr. 3, pp. 447–460

[LuCR01] Luo, M. R., G. Cui, and B. Rigg: The development of the CIE 2000 colour-difference formula: CIEDE2000. In: *Color Research and Application* vol. 26 (2001), pp. 340–350

[Mari13] Mariescu-Istodor, R.: *"Detecting user actions in MOPSI"*, University of Eastern Finland, 2013

[MTSF14] Mariescu-Istodor, R., A. Tabarcea, R. Saeidi, and P. Fränti: "Low Complexity Spatial Similarity Measure of GPS Trajectories". In: *WEBIST 2014*. Barcelona, Spain, 2014, pp. 62–69

[MCSL05] Markowetz, A., Y.-Y. Chen, T. Suel, X. Long, and B. Seeger: "Design and Implementation of a Geographic Search Engine". In: *Eighth International Workshop on the Web and Databases (WebDB 2005)*. Baltimore, USA, 2005

[MMSK08] Matyas, S., C. Matyas, C. Schlieder, P. Kiefer, H. Mitarai, and M. Kamata: "Designing location-based mobile games with a purpose: collecting geospatial data with CityExplorer". In: *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*. Yokohama, Japan, 2008, pp. 244–247

[Mccu01] McCurley, K. S.: "Geospatial mapping and navigation of the web". In: *Proceedings of the 10th international conference on World Wide Web*. vol. WWW '01: P. Hong Kong, 2001, pp. 221–229

[MiMG99] Mikheev, A., M. Moens, and C. Grover: "Named Entity recognition without gazetteers". In: *Proceedings of the 9th conference on European chapter of the Association for Computational Linguistics*. Bergen, Norway, 1999, pp. 1–8

[MHKT09] Misund, G., H. Holone, J. Karlsen, and H. Tolsby: "Chase and Catch - simple as that?". In: *Proceedings of the International Conference on Advances in Computer Enterntainment Technology - ACE '09*. Athens, Greece, 2009, p. 73

[NaRa02] Navarro, G. and M. Raffinot: *Flexible pattern matching in strings: practical on-line search algorithms for texts and biological sequences* : Cambridge University Press, 2002

[NeJu12] Neustaedter, C. and T. K. Judge: "See it". In: *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion - CSCW '12*. New York, New York, USA, 2012, p. 235

[PBMW99] Page, L., S. Brin, R. Motwani, and T. Winograd: The PageRank citation ranking: Bringing order to the web. In: *Technical report, Stanford Digital Library Technologies Project*, Stanford InfoLab (1999)

[PaMP03] Patterson, C. A., R. R. Muntz, and C. M. Pancake: Challenges in location-aware computing. In: *Pervasive Computing, IEEE* vol. 2, IEEE (2003), Nr. 2, pp. 80–89

[PCRC08] Petit, M., C. Claramunt, C. Ray, and G. Calvary: A design process for the development of an interactive and adaptive GIS. In: *Lecture Notes in Computer Science* vol. 5573 (2008), Nr. Web and Wireless Geographical Information Systems, pp. 96–106

[PiTh02] Piekarski, W. and B. Thomas: ARQuake: the outdoor augmented reality gaming system. In: *Communications of the ACM* vol. 45 (2002), pp. 36–38

[PCJA07] Purves, R. S., P. Clough, C. B. Jones, A. Arampatzis, B. Bucher, D. Finch, G. Fu, H. Joho, et al.: The design and implementation of SPIRIT: a spatially aware search engine for information retrieval on the Internet. In: *International Journal of Geographical Information Science* vol. 21, Taylor & Francis (2007), Nr. 7, pp. 717–745

[QXFX10] Qin, T., R. Xiao, L. Fang, X. Xie, and L. Zhang: "An efficient location extraction algorithm by leveraging web contextual information". In: *Proceedings of the 18th*

SIGSPATIAL international conference on advances in geographic information systems. San Jose, CA, USA, 2010, pp. 53–60

[RMCE06] Rashid, O., I. Mullins, P. Coulton, and R. Edwards: Extending cyberspace: location based games using cellular phones. In: *Computer Entertainment* vol. 4 (2006), p. 4

[ScVo04] Schiller, J. and A. Voisard: *Location Based Services.* 1st. ed. : Morgan Kaufmann Publishers Inc., 2004

[ScKM05] Schlieder, C., P. Kiefer, and S. Matyas: "Geogames: A conceptual framework and tool for the design of location-based games from classic board games". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. vol. 3814 LNAI, 2005, pp. 164–173

[SMRS13] Schmidt, S., S. Manschitz, C. Rensing, and R. Steinmetz: "Extraction of Address Data from Unstructured Text using Free Knowledge Resources". In: *Proceedings of the 13th International Conference on Knowledge Management and Knowledge Technologies*. Graz, Austria, 2013, p. 7

[SeBD09] Setlur, V., A. Battestini, and X. Ding: "Travel scrapbooks: Creating rich visual travel narratives". In: *2009 IEEE International Conference on Multimedia and Expo*. New York, NY, USA : IEEE, 2009, pp. 1314–1317

[ShBa11] Shi, G. and K. Barker: "Extraction of geospatial information on the Web for GIS applications". In: *IEEE 10th International Conference on Cognitive Informatics and Cognitive Computing (ICCI-CC'11)*. Banff, AB, Canada, 2011, pp. 41–48

[SMCA06] Silva, M. J., B. Martins, M. Chaves, A. P. Afonso, and N. Cardoso: Adding geographic scopes to web resources. In: *Computers, Environment and Urban Systems* vol. 30 (2006), pp. 378–399

[SDBD05] Souza, L. A., C. A. Davis Jr., K. A. V. Borges, T. M. Delboni, and A. H. F. Laender: "The Role of Gazetteers in

Geographic Knowledge Discovery on the Web". In: *Third Latin American Web Congress (LA-WEB'2005)*. vol. 2005. Buenos AIres, Argentina : IEEE, 2005, pp. 157–165

[SpMi08] Spikol, D. and M. Milrad: "Combining physical activities and mobile games to promote novel learning practices". In: *Proceedings - 5th IEEE International Conference on Wireless, Mobile, and Ubiquitous Technologies in Education, WMUTE 2008*. Beijing, China, 2008, pp. 31–38

[SuKo06] Suomela, R. and A. Koivisto: "My photos are my bullets–using camera as the primary means of player-to-player interaction in a mobile multiplayer game". In: *Entertainment Computing-ICEC 2006*. Cambridge, UK, 2006, pp. 250–261

[TaFM09] Tabarcea, A., P. Fränti, and V. Manta: Using a Spatial Database in a Location-Based Search Application. In: *Buletinul Institutului Politehnic Iasi* vol. LV (LIX) (2009), Nr. 3, pp. 55–62

[Tsai11] Tsai, F. S.: Web-based geographic search engine for location-aware search in Singapore. In: *Expert Systems with Applications* vol. 38, Elsevier (2011), Nr. 1, pp. 1011–1016

[WTCF12] Waga, K., A. Tabarcea, M. Chen, and P. Fränti: "Detecting Movement Type by Route Segmentation and Classification". In: *Conference on Collaborative Computing: Networking, Applications and Worksharing*. Pittsburgh, Pennsylvania, United States, 2012, pp. 508–513

[WaTF12] Waga, K., A. Tabarcea, and P. Fränti: "Recommendation of points of interest from user generated data collection". In: *CollaborateCom*. Pittsburgh, Pennsylvania, United States, 2012, pp. 550–555

[WTMF13] Waga, K., A. Tabarcea, R. Mariescu-Istodor, and P. Fränti: "Real Time Access to Multiple GPS Tracks.". In: *WEBIST*. Achen, Germany, 2013, pp. 293–299

[Wan14] Wan, Z.: *"O-Mopsi: Location-based Orienteering Mobile Game"*, University of Eastern Finland, 2014

[WXWL05] Wang, C., X. Xie, L. Wang, Y. Lu, and W.-Y. Y. Ma: "Detecting geographic locations from Web resources". In: YORK:ACM, N. (ed.): *Proceedings of the 2005 Workshop in Geographic Information Retrieval*. Bremen, Germany, 2005, pp. 17–24

[WaAm03] Watters, C. and G. Amoudi: GeoSearcher: Location-based ranking of search engine results. In: *Journal of the American Society for Information Science and Technology* vol. 54, Wiley Online Library (2003), Nr. 2, pp. 140–151

[WeBO12] Wetzel, R., L. Blum, and L. Oppermann: "Tidy city". In: *Proceedings of the International Conference on the Foundations of Digital Games - FDG '12*. Raleigh, North Carolina, USA, 2012, p. 238

[ViNa05] Viola, P. and M. Narasimhan: "Learning to extract information from semi-structured text using a discriminative context free grammar". In: *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. vol. 272, 2005, pp. 330–337

[Väns04] Vänskä, I.: *"Using location information in web documents"*, University of Eastern Finland, 2004

[YoTM01] Yokoji, S., K. Takahashi, and N. Miura: Kokono Search: A Location Based Search Engine. In: *10th International World Wide Web Conference (WWW10)*. Raleigh, North Carolina, USA (2001)

[YKSJ09] Yu, Y. H., J. H. Kim, K. Shin, and G. S. Jo: Recommendation system using location-based ontology on wireless internet: An example of collective intelligence by using "mashup" applications. In: *Expert Systems with Applications* vol. 36 (2009), pp. 11675–11681

[ZJLY12] Zhang, Q., P. Jin, S. Lin, and L. Yue: "Extracting focused locations for web pages". In: *Web-Age Information Management* : Springer, 2012, pp. 76–89

Andrei-Cătălin Tabarcea

Location-Based Web Search and Mobile Applications

Due to the rapid development and wide availability of positioning techniques and internet connectivity, location is easily available and significantly improves the applications that utilize the user's context. This thesis presents contributions in the field of location-based services. It proposes applications and advances in location-based web search and data mining, postal address detection and location-based gaming using data collected by users within the MOPSI project.

UNIVERSITY OF
EASTERN FINLAND

Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences

ISBN: 978-952-61-1868-0 (printed)
ISBN: 978-952-61-1869-7 (pdf)
ISSNL: 1798-5668
ISSN: 1798-5668
ISSN: 1798-5676 (pdf)