The Islamic University–Gaza

Research and Postgraduate Affairs

Faculty of Information Technology

Master of Information Technology

الجـامعـــــــــة الإســـــلاميـة – غـزة

شئون البحث العلمي والدراسات العليا

كليــــــة تكنولوجيا المعلومات

ماجستير تكنولوجيا المعلومات

A Mobile Context-Aware Recommendation System Based on DBpedia

نظام اقتراح متنقل معتمد على السياق والديبيبيديا

Mohmoud S. A. Al Halabi

Supervised by

Dr. Iyad M. AlAgha

Assistant Professor of Computer Science

A thesis submitted in partial fulfilment of the requirements for the degree of

Master of Information Technology

March/2017

إقـــــرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

نظام اقتراح متنقل معتمد على السياق والديبيبيديا

A Mobile Context-Aware Recommendation System Based on DBpedia

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وأن هذه الرسالة ككل أو أي جزء منها لم يقدم من قبل الآخرين لنيل درجة أو لقب علمي أو بحثي لدى أي مؤسسة تعليمية أو بحثية أخرى. وأن حقوق النشر محفوظة للجامعة الإسلامية – غزة.

Declaration

I hereby certify that this submission is the result of my own work, except where otherwise acknowledged, and that this thesis (or any part of it) has not been submitted for a higher degree or quantification to any other university or institution. All copyrights are reserves to IUG.

Student's name:	محمود سليمان عبد الرحمن الحلبي	اسم الطالب:
Signature:		التوقيع:
Date:	2017/05/14م	التاريخ:

نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة شئون البحث العلمي والدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ **محمود سليمان عبدالرحمن الحلبي** لنيل درجة الماجستير في كلية *تكنولوجيا المعلومات* برنامج <u>تكنولوجيا المعلومات</u> وموضوعها:

نظام اقتراح متنقل معتمد على السياق والديبيبيديا

A Mobile Context-Aware Recommendation System Based on DBpedia

وبعد المناقشة التي تمت اليوم الأربعاء 14 رجب 1438هـ، الموافق 2017/04/11م الساعة الواحدة ظهراً، في قاعة مؤتمرات القدس، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

د. إيـــاد محمـــد الأغـــا	مشـرفاً و رئيسـاً	....
د. توفيـــق ســليمان برهـــوم	مناقشــاً داخليــاً	....
د. نـــاجي شــكري الظاظـا	مناقشـاً خارجيـاً	....

وبعد المداولة أوصت اللجنة بمنح الباحث درجة الماجستير في كلية *تكنولوجيا المعلومات*/ برنامج **تكنولوجيا المعلومات**.

واللجنة إذ تمنحه هذه الدرجة فإنها توصيه بتقوى الله ولزوم طاعته وأن يسخر علمه في خدمة دينه ووطنه.

والله ولي التوفيق،،،

نائب الرئيس لشئون البحث العلمي والدراسات العليا

أ.د. عبدالرؤوف علي المناعمة

Abstract

Context-aware recommendation systems generate more relevant recommendations by adapting them to the specific contextual situation of the user. They have been widely used for tourism to recommend locations relevant to the user's needs. A plenty of efforts have presented different approaches to capture the user's needs and contextual information, and then suggest tourist locations that fulfill the user's requirements. These approaches, however, often rely on conventional mappings services, like Google maps, to identify locations and their corresponding categories (e.g. restaurants, libraries, shopping, etc). For example, if the user is looking for museums, the recommendation system will refer to the locations' database and categorization offered by the mapping service. Unless the mapping service provides sufficient information about locations on the map, recommendation approaches will not be able to identify target locations. In this work, we proposed a context-aware recommendation approach that leverages Linked Open Data (LOD), and DBpedia in particular, to provide recommendations that are semantically related to the user's needs. The proposed approach prompts the user to input a search query (i.e. keywords to search for locations of interest). Then, it will use the user's GPS location and keywords to query DBpedia for locations that best match the user's interests. The proposed approach has three contributions: First, it can offer recommendations of geographical locations not covered by traditional mapping services. Our approach does not use the location's database of the mapping service. Instead, it seeks to extract location's details by directly querying DBpedia. Second, it uses the search keywords submitted by the user to search DBpedia for locations that best match the user's needs. Third, it presents an algorithm that ranks the recommendations in a way that balances between proximity (i.e. distance to the user) and relevance to the user's interests. A prototype mobile application was also developed to demonstrate the use of the proposed recommendation approach. The performance and efficiency of the proposed approach was assessed by using a dataset of 41 queries covering three cities. We used two evaluation metrics to assess the performance and the ranking of results. The system achieved 77.77% using the MAP metric with SD of 0.029, and 91.377% Normalized Discount Cumulative Gain with SD 0.337, and average of 6.27 second in query execution with 2.5 SD. The archived results as shown in the previous values indicates that nDCG value was greater than the MAP value. This result proves that the system achieved better results in ranking than in generating accurate results.

Keywords: Recommendation System, semantic search, user context, LOD, DBpedia.

الملخص

تولد انظمة الإقتراح المعتمدة على السياق الخاص بالمستخدم (Context–aware recommendation systems) توصيات اكثر صله بالمستخدم بناء على حالة المستخدم (user context). يستخدم هذا النوع من المقترحات على نطاق واسع في اقتراح اماكن سياحية تلبي احتياجات المستخدم. الكثير من الجهود بذلت بطرق عدة لتحديد احتياجات المستخدم وظروفه , واستخدمت هذه البيانات لإقتراح مواقع سياحية تلبي احتياجات المستخدم. تعتمد هذه الطرق غلى خدمات الخرائط التقليدية, مثل خرائط جوجل, لتحديد الأماكن وتصنيفها (علي سبيل المثال، مطاعم، مكتبات، اماكن تسوق إلخ). فمثلا , إذا كان المستخدم يبحث عن متحف سيرجع النظام إلى قاعدة بيانات الموقع والفئات المصنفة مسبقاً في خدمات الخرائط. و إذا لم يتوفر معلومات عن هذا الموقع على خدمات الخرائط، فلن يستطيع نظام الاقتراح تحديد الموقع المطلوب. في هذا العمل، قدمنا نظام إقتراح يراعي الظروف الانية للمستخدم ويستفيد من قواعد البيانات المفتوحة ((Linked Open Data (LOD)، وبالذات خدمات DBpedia، وذلك بهدف توفير إقتراحات ذات صلة دلالية بإحتياجات المستخدم. العمل المقترح يطلب من المستخدم إدخال إستعلام بحثي (مثل كلمات دلالية للبحث عن المكان المهتم به). من ثم سيقوم النظام بإستخدام إحداثيات موقع المستخدم GPS والكلمات المفتاحية للإستعلام في DBpedia عن المواقع الأكثر توافقاً مع إهتمامات المستخدم. تساهم الطريقة المقترحة بثلاث إضافات: الإضافة الأولى، يمكنه تقديم إقتراحات حول اماكن جغرافية لم يتم تغطيتها من قبل خدمات الخرائط التقليدية. فالطريقة المتبعة لا تستخدم قاعدة بيانات المواقع الخاصة بخدمات الخرائط. وبدلاً من ذلك تسعي لإستخراج معلومات الموقع بالإستعلام المباشر DBpedia. الإضافة الثانية: يستخدم النظام كلمات البحث المدخلة من قبل المستخدم للبحث في DBpedia عن الموقع الأكثر تلبية لاحتياجات المستخدم. الإضافة الثالثة : يقدم خوارزمية تقوم بترتيب النتائج بطريقة توازن بين البعد الجغرافي للموقع عن المستخدم ومدي صلته بإهتمام المستخدم. حيث تم تطوير نموذج مبدئي لتطبيق الطريقة وذلك بهدف عرض إستخدام طريقة الإقتراح المقدمة. تم تقييم اداء وكفاءة الطريقة المقترحة بإستخدام مجموعة بيانات مكونة من 41 إستعلام تشمل ثلاثة مدن، وإستخدمنا مقياسي تقييم بهدف إختبار اداء النظام وتقييم ترتيب النتائج. حقق النظام 77.77% بإستخدام مقياس MAP بإنحراف معياري 0.029، و 91.377% بإستخدام معيار تطبيع الكسب التراكمي (Normalized Discount Cumulative Gain) بإنحراف معياري 0.337، ومتوسط زمن إستعلام 6.27 ثانية بإنحراف معياري 2.5. تظهر النتائج أعلاه أن قيم nDCG أعلي من القيم الناتجه في فحص MAP مما يدلل على ان النظام حقق نتائج في عملية ترتيب الإقتراحات أفضل من النتائج المحققة في دقة الإقتراحات.

بسم الله الرحمن الرحيم

قال تعالى : {يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ}

[المجادلة: 11]

وقال تعالى: {شَهِدَ اللَّهُ أَنَّهُ لاَ إِلَـهَ إِلاَّ هُوَ وَالْمَلاَئِكَةُ وَأُوْلُواْ الْعِلْمِ قَآئِمَاً بِالْقِسْطِ}

[آل عمران: 18]

Dedication

To those whose kindness, patience and support were the candles that enlightened my path towards success; my Father and Mother.

To my beloved wife who saved no efforts in encouraging and supporting me during my journey toward success, and her mother for the continues support she provides.

To my brothers who saved no effort supporting me. specially my brother Montaser

To my friends who urged me to complete the task.

Acknowledgment

My gratitude is deeply paid to my advisor, Dr. Iyad M. AlAgha for his generosity, guidance and advice. Of course, I would not to forget the discussion committee for accepting to discuss this thesis for defence.

Special thanks are extended to IUGaza specially the faculty of IT for providing the opportunity to this achievement .

My gratitude is extended to all my True Friends and Colleagues who provided me with every help they can.

Table of Contents

IX

List of Tables

List of Figures

List of Abbreviation

LOD	Linked Open Data
NLP	Natural Language Processing
RDF	Resource Description Framework
SPARQL	Simple Protocol and Resource Description Framework Query Language
MAP	Mean Average Precision
nDCG	Normalized Discount Cumulative Gain
SD	Standard Deviation
JSON	JavaScript Object Notation
URL	Uniform Resource Locator
POI	Points Of Interests

Chapter 1
Introduction

Chapter 1   Introduction

1.1   Introduction

Nowadays the amount of publicly available data on the World Wide Web has dramatically increased and has led to a problem of information overload. The recommendation systems try to tackle this issue by offering personalized suggestions. A recommendation system uses mechanisms to predict associated items and provide the information to the user based on criteria such as user preferences, item popularity or demographic factors. The recommendation system also plays a vital role in mobile websites browsing in order to overcome limitations of mobile device due to restricted user interface, screen size, connectivity and information overload (Paireekreng, 2013). Personalized recommendation systems recommend items, which an end user prefers by using automatic information filtering method. Moreover, as mobile computing progresses, various resources can be available to model user preference. A Mobile device provides a user with information and services related to the physical location based on user's location. since there are lots of information and services, it is difficult to find a proper services of the end preference at the proper time (Park, Hong, & Cho, 2007).

Recommendation systems collect information from the user mobile phone, From a large information Sources the Recommender carefully chooses suggested information and suggest individual information to each. The recommendation system uses content-based filtering and/or collaborative filtering (Bouneffouf, Bouzeghoub, & Gançarski, 2012). Some research works tried to take the user's position into consideration while making a recommendation.

Context-aware recommendation systems generate more relevant recommendations by adapting them to the specific contextual situation of the user (Danylenko & Löwe, 2012). For example, the system can recognize the user's location using GPS, and then provide recommendations related to the user's location. Context-aware recommendation systems can exploit different types of contextual information such as location, atmosphere, acoustics and light.

Context-aware recommendation systems have been widely used for tourism to recommend locations relevant to the user's needs. A plenty of efforts have presented different approaches to capture the user's needs and contextual information, and then suggest tourist locations that fulfill the user's requirements (Gavalas, Konstantopoulos, Mastakas, & Pantziou, 2014). These approaches, however, often rely on conventional mappings services, like Google maps, to identify locations and their corresponding

categories (e.g. restaurants, libraries, shopping, etc). For example, if the user is looking for museums, the recommendation system will refer to the locations' database and categorization offered by the mapping service. Unless the mapping service provides sufficient information about locations on the map, recommendation approaches will not be able to identify target locations.

DBpedia is defined in the official DBpedia community web site ("DBpedia," 2017) as a crowd-sourced community effort which goal is to extract information from the Wikipedia and organize it in a structured information, and make this information available on the Web. This structure will makes it possible to run sophisticated queries on Wikipedia and DBpedia, which allows developers to link the different data sets on the Web to Wikipedia data.

in order to represent metadata about WWW resources the Resource Description Framework (RDF) was developed. the RDF makes it possible to represent the URI resources on WWW in graph form, furthermore the RDF uses XLS which allows easy exchange of information among networks of heterogeneous computer (Pan & Horrocks, 2007).

SPARQL it is an acronym of Protocol and RDF Query Language, its main function in run query on ontologies, those quires can retrieve and update information on the ontologies, similar to the traditional DB SQL, SPARQL have many commands and statements (El-Radie & Alagha, 2015).

In this work, we proposed a context-aware recommendation approach that leverages Linked Open Data (LOD), and DBpedia in particular, to provide recommendations that are semantically related to the user's needs. The proposed approach prompts the user to input a search query (i.e. keywords to search for locations of interest). Then, the approach will use the user's GPS location and keywords to query DBpedia for locations that best match the user's interests.

The proposed approach has three contributions:

First, it can offer recommendations of geographical locations not covered by traditional mapping services. Our approach does not use the location's database of the mapping service. Instead, it seeks to extract location's details by directly querying DBpedia.

Second, it uses the search keywords submitted by the user to search DBpedia for locations that best match the user's needs.

Third, it presents an algorithm that ranks the recommendations in a way that balances between proximity (i.e. distance to the user) and relevance to the user's interests.

A prototype mobile application was also developed to demonstrate the use of the proposed recommendation approach. The Content based recommendation system (Ostuni, Di Noia, Mirizzi, Romito, & Di Sciascio, 2012) which uses Linked Open Data LOD-enabled content-based from the freely available semantic datasets caused the datasets to boom . These information, which are encoded to be understandable by machine and have RDF triples can be manipulated to represent items and build a user profile in a Linked Open Data LOD-enabled content-based. The availability of various data related to various knowledge domains were made available based on using the dataset. Using the DBpedia datasets, we will be able to access rich linked data mapped to a huge variety of topics. All of this is made available because of the SPARQL endpoints.

1.2 Statement of the Problem

The problem investigated in this work is two-fold:

First, many geographical areas in the world are not covered in detail by common mapping services. Therefore, there is a need for an alternative solution for areas not covered in traditional mapping services.

Second, traditional mapping services often depend on predefined classifications and existing information of locations. Therefore, there is a need for a recommendation approach that do not rely on predefined knowledge of locations on map.

1.3 Research questions

1. How to exploit LOD, such as DBpedia as an extension to traditional mapping services in order to recommend locations relevant to the user's needs?
2. How to rank recommended locations by making the best balance between the proximity of locations and their relevance?
3. How to integrate the proposed recommendation approach into a mobile application that can be easily used by users in practice?

1.4 Objectives

1.4.1 Main Objective

The main objective is to design a context-based recommendation approach that utilizes DBpedia, rather than mapping services, to identify locations relevant to the user needs. The relevance of recommendations is determined in terms of proximity and semantic relatedness.

1.4.2 Specific Objectives:
The main objective can be split into the following specific objectives:

- Explore how LOD, and DBpedia in particular, can be queried for details of locations. The aim is to use LOD as an extension to conventional mapping services when locations are not covered on maps.
- Develop an algorithm to identify and rank recommended locations based on both the proximity to user's location and the semantic relatedness of locations to the user's needs.
- Design a prototype mobile application that uses the proposed approach to recommend locations to users in practice.
- Investigate the appropriate evaluation metrics to assess the performance of the proposed approach so that both the accuracy and ranking of results are assessed.

1.5   Importance of Research
Context-aware recommendation systems have gained a considerable attention in recent years, and plenty of works have explored a variety of approach to recommend items (e.g. locations and news) relevant to the user needs. The importance of this work is that:

- Extending previous efforts by proposing an approach that can identify locations that may not be covered by conventional mapping services.
- This approach exploits the recent advances in LOD in order to effectively determine the importance and relevance of recommended locations.

1.6   Scope and Limitations of the Project
- The proposed approach is relevant only when locations are not properly covered by common mapping services. It does not aim to replace mapping services, but to extend them to improve the coverage of locations.
- The evaluation process focused on the assessment of the underlying recommendation approach by using relevant metrics. The usability of the applications was not evaluated as it is beyond our objectives.
- The prototype mobile application was developed on Android only. It also assumes that the mobile device is connect to the Internet while using the recommendation service.
- The evaluation result, are not compared to results of other approaches because we could not find other comparable approaches to ours.

1.7   Overview of Thesis
This thesis document is organized into five chapters as follow:

Chapter 1: Introduction: This chapter presents an overview of the main problem and possible solutions and focuses on proposed solution.

Chapter 2: Related Works: This chapter focuses on related works about various recommendation systems and their design.

Chapter 3: Methodology: This chapter explains in detail the steps followed to build our recommendation system.

Chapter 4: Evaluation: This chapter presents the evaluation process, and describes the performed tests, and discusses the results.

Chapter 5: Conclusion: This chapter presents a conclusion of this thesis and discusses future works.

Chapter 2
Literature Review

Chapter 2   Literature Review

2.1 Background

This chapter briefly explains the background of this work, including an overview on recommendation systems and its types. LOD and DBpedia will be also introduced, focusing on the use of DBpedia for information retrieval systems. Afterwards, related works will be reviewed and discussed.

2.1.1 Recommendation Systems

The recommendation systems can be named as recommendation platforms or recommendation engines. Their main objective is to predict information relevant to the user's needs according to some rating or preference (Ricci, Rokach, & Shapira, 2011; Sridevi, Rao, & Rao, 2016). Due to the huge amount of the available online data, making a good sensible decision is a main challenge facing the users. A recommendation system is a piece of software program designed to help individual internet users to better deal with this issue and help them make a better decision.

Recommendation systems have become extremely common in recent years. They are commonly used in various application areas. Some common applications include selecting research articles, search queries, movies, and products in general. Some recommendation systems are designed for experts (Chen, Ororbia, Alexander, & Giles, 2015) collaborators,(Chen, Gou, Zhang, & Giles, 2011) jokes, restaurants, garments, and financial services(Felfernig, Isak, Szabo, & Zachar, 2007).

Typical recommendation systems produce a list of recommendations based on one of the two main approaches which are:

1) Collaborative filtering.

2) content-based filtering.

These two approaches are knows as the personality-based approach (Jafarkarimi, Sim, & Saadatdoost, 2012). Filtering approaches rely on the user's past actions such as previously visited areas, purchased items and the rating the user granted to some areas and products.

Collaborative filtering uses similar decisions made by other users. This model is then used to predict and suggest places that the user may have an interest in (Melville & Sindhwani, 2010). In order to recommend additional items with similar properties, the content-based filtering approaches utilize a series of discrete characteristics of an item

(Mooney & Roy, 2000). These approaches are often combined in a Hybrid fashion recommendation Systems (Mooney & Roy, 2000).

From an intelligent agent perspective. Montaner et al. (Montaner, López, & De La Rosa, 2003) provided the first overview of recommendation systems. They have analyzed 37 different recommendation systems and their references, and sorted them into a list of 8 basic dimensions. These dimensions are then used to establish a taxonomy under which the systems analyzed are classified. At the end they concluded by a cross-dimensional analysis with the aim of providing a starting point for researchers to construct their own recommendation system.

Adomavicius and Tuzhilin (Adomavicius & Tuzhilin, 2005) provided a new, alternate overview of recommendation systems. They described various limitations of current recommendation methods and suggested possible extensions that can improve recommendation capabilities in order to make recommendation systems applicable in broader range of applications. The extensions they suggested include: improvement of understanding of users and items, using of contextual information in the recommendation process, and using multi-criteria ratings.

2.1.2 Types of Recommendation Systems

According to the literature, recommendation systems can be classified into three main categories. A content-based filtering, collaborative filtering and hybrid approaches (BRAMSTÅNG & JIN, 2015). These categories are explained in detail in the following subsections.

2.1.2.1 Content-Based filtering

Content-based filtering is a common recommendation approach that uses the description of an item and a profile of the user's preference. Keywords are used to describe the items and a user profile. Those keywords are built to indicate the type of item this user likes. I cannot understand the role of keywords: give me an example. The content-based filtering algorithms try to recommend items that are similar to those that a user liked in the past (or is examining in the present). Various candidate items are compared with items previously rated by the user and the best-matching items are recommended (Ghazanfar, Prügel-Bennett, & Szedmak, 2012) . The context based recommendation system can be adopted in mobile recommendation systems by using a set of variables including the GPS location, surrounding places, popularity of a place in the area plus the user interests. Some of the researches and applications using the this approach are : (Ricci & Nguyen, 2007) (Bouneffouf et al., 2012) and (Felfernig et al., 2007).

2.1.2.2 Collaborative filtering

Collaborative filtering works on an opposite approach of content-based filtering approach. The Collaborative filtering uses similarities between users instead of similarities between items. The main scheme is to group users with similar behavior, e.g. purchases. These groups are used as a source for recommendations. The moment a user visits a product, an association of the user can be made with one or more groups of users who have purchased that product, and recommendations can be produced based on what these other users have purchased beside the current product (BRAMSTÅNG & JIN, 2015). Collaborative approaches usually suffer from data lack and the cold start problem(Adomavicius & Tuzhilin, 2005). The cold start problem is caused when a lot of users buy very different products, thus making it difficult to create groups of users with similar behavior. Cold start problem can also happen when a user visits a product that no one has bought. No users with similar behaviors can be found because they are not there. A good point for using the Collaborative filtering is that when a big amount of data is available, the recommendations become reliable because they reflect actual user behavior (Adomavicius & Tuzhilin, 2005). Some of the researches and applications using the this approach are :(V. W. Zheng, Zheng, Xie, & Yang, 2010) (Y. Zheng, Zheng, & Xie, 2014) and (de Spindler, Norrie, Grossniklaus, & Signer, 2006).

2.1.2.3 Hybrid Approaches

The hybrid approach is a combination of the content-based and a collaborative approach. The objective is to overcome the disadvantages of each method and utilize the best of them. Similarities between items are used to bypass the cold start problem. When a cold start is about to happen the system uses the content based approach and detect similarities between products. Bothe approaches are combined in deferent ways depending on the targeted problem (BRAMSTÅNG & JIN, 2015). Some of the researches and applications using the this approach are : (Liu & Shih, 2005) (Ziegler, Lausen, & Schmidt-Thieme, 2004) and (Hjortdal, Redington, De Leval, & Tsang, 2002).

The proposed mobile-based recommendation approach in this these uses content based recommendation in order to identify DBpedia resources that best relate to the user needs. The approach also aims to make balance between content and context based recommendations by recommending proximate locations that fulfil the user's needs.

2.1.3 DBpedia and its Uses in Recommendation System

The DBpedia Ontology organizes the knowledge on Wikipedia in 320 classes which form a hierarchy and are described by 1,650 different properties. It features labels and abstracts for 3.64 million things in up to 97 different languages of which 1.83 million are classified in a consistent ontology, including 416,000 persons, 526,000

places, 106,000 music albums, 60,000 films, 17,500 video games, 169,000 organizations, 183,000 species and 5,400 diseases. Additionally, there are 6,300,000 links to external web pages, 2,724,000 links to images, 740,000 Wikipedia categories and 690,000 geographic coordinates for places (Mendes, Jakob, & Bizer, 2012). The DBpedia dataset contains information about almost 300,000 locations. DBpedia data about these locations is interlinked with various other location related datasets, such as the GeoNames, US Census, CIA Factbook, and EuroStat datasets. Altogether there are around 185,000 external RDF links into other RDF datasets on the Web, making DBpedia an important interlinking hub (Becker & Bizer, 2008). DBpedia is a community effort to extract structured information from Wikipedia and to make this information available on the Web. DBpedia allows you to ask sophisticated queries against datasets derived from Wikipedia and to link other datasets on the Web to Wikipedia data (Auer et al., 2007). RDF is a directed, labeled graph data format for representing information in the Web. RDF is often used to represent, among other things, personal information, social networks, metadata about digital artifacts, as well as to provide a means of integration over disparate sources of information. This specification defines the syntax and semantics of the SPARQL query language for RDF (Prud'hommeaux & Seaborne, 2006).

According to the official website of the DBpedia community ("DBpedia," 2017) the "DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web. Makes it possible to run sophisticated queries on Wikipedia and DBpedia, which allows developers to ask sophisticated queries against Wikipedia, and to link the different data sets on the Web to Wikipedia data."

DBpedia is extracted from Wikipedia as shown in Fig. 2.1: Various sets of structured information in multiple languages are extracted from Wikipedia by using open source extraction frameworks. These extractions are RDF triples and are added to the knowledge base as properties of the corresponding URI (Bizer et al., 2009). The structured content of DBpedia enables computer systems to capture relations between topics and resources. On the other hand, data in Wikipedia is non structured, thus cannot be directly used by computer systems. (wikipedia, 2017).
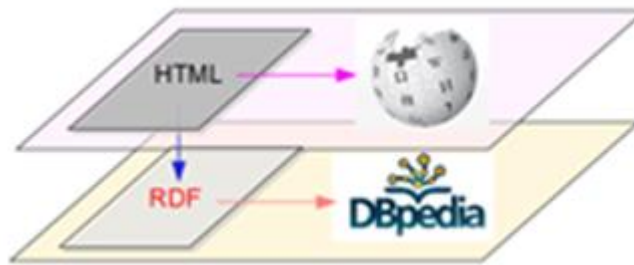
Figure (2.1): illustrate DBpedia extract Wikipedia ("ibm," 2016)

Figure (2.2): illustrate DBpedia extract Wikipedia (Corlosquet & Clark, 2011)

The DBpedia Ontology as in Fig. 2.2 organizes the knowledge on Wikipedia in 320 classes which form a sub assumption hierarchy and are described by 1,650 different properties (Mendes et al., 2012). The DBpedia dataset contains information about almost 300,000 locations (Becker & Bizer, 2008). This specification defines the syntax and semantics of the SPARQL query language for RDF (Prud'hommeaux & Seaborne, 2006).

2.1.4 Using DBpedia in Recommendation Systems

Every recommendation system requires access to a rich source of information, where the system will get the information and filter them according to the user interest and location. One of the central knowledge source for humans is Wikipedia, Wikipedia

is maintained and enriched by thousands of contributors. Most of Wikipedia articles are built from natural languages text; other structured information exists in the Wikipedia as well. These structured information consist of info-box templates, categorization information, images, geo-coordinates, and links to external Web pages . Other projects like DBpedia extracts from Wikipedia various sets of structured information in multiple languages using an open source extraction framework. This extraction framework combines and mixes all Wikipedia information into multilingual, multi-domain knowledge base. Uniform Resource Identifier URI for every page in Wikipedia is created and inserted in the DBpedia, this URI is used to identify a concept or Identity an entity described the corresponding Wikipedia Page, during the extraction process is performed by the framework, in the process structured information from the wiki such as categories, page links and infobox fields are extracted. These extraction are RDF triples and are added to the knowledge base as properties of the corresponding URI (Bizer et al., 2009).

Some of the researches and applications have used DBpedia as information source for recommendation system. Examples of these application include Cinammapy, which relies on the RDF graph of the DBpedia to find the similarity between two movies or two resources weather they are directly related or are the subject of two RDF triples share the same property and the same object, or the object of two RDF triples having the same property and the same subject. (Ostuni et al., 2012). The dbrec is a music recommendation system built on top of DBpedia, in the work the developer built a Linked Data Semantic Distance (LDSD) algorithm, which computes the semantic distance between two objects of DBpedia ontology. This algorithm will be applies on the identified relevant subset fetched from DBpedia, this subset have been reduced for query optimization. The dbrec relies on the calculated semantic distance in order to make recommendations (Passant, 2010). Proposed a graph-based recommender system, the features are automatically extracted from the Linked Data and feed to the system. In most of the cases, the used techniques require additional information from the user in order to generate better recommendations. because this proposed recommendation system relies only on Linked Data it does not require reducing the Dataset resources and links belonging to a specific domain (Musto et al., 2015).

Our work relies on DBpedia as information source in particular, to retrieve content and geo-locations, it extracts location's details by directly querying DBpedia using SPARQL. The SPARQL Query consists of the user GPS location later a ranking algorithm finds the relevance degree with to user search keyword, and balances the relevance with the distance between the user location and the place location. The final recommendation given to the user is based on the balancing algorithm.

2.1.5 RDF and RDFS

Resource Description Framework (RDF) is a framework for representing information about resources in a graph form. It has been developed to represent metadata about WWW resources with Uniform Resource Identifier (URI). XML is the language used in writing RDF documents, a version of it called RDF/XML is used in writing RDF documents. The use of XML allows easy exchange of RDF information among heterogeneous computer machines using deferent types of operating systems and programming languages. Triples of subject-predicate-object are used to represent information in RDF. Except the last element of the triple represent resources. The last element is called a literal; it is a constant like a string or a number. RDF have can have many ontologies. These ontologies are defined using Web Ontology Language (WOL), this many ontologies are expressed some times by having RDF schema (RDFS) system. The RDFS provides mechanisms for describing group of related resources and relationships between these resources (Pan & Horrocks, 2007).

2.1.6 Ontology

Ontology is a derived from Greek word "onto" meaning "being", the suffix ology means sciences, this give the meaning of ontology and the science of being. (Bäck, Vainikainen, Södergård, & Juhola, 2003) Genesereth and Nilsson defined Ontology as an explicit specification of a set of objects. concepts, and other entities which are presumed to exist in some area of interest and the relationships that holding them. Ontology provides a shared a clear conceptual hierarchy and strong logical sequences. Ontology contains a set of clearly described specific items including classes/concepts including their concepts, slot, restriction, facet and a series of instance related to one class, which combines to the knowledge storage. The Core of ontology is Class, which describes the concepts in a domain. The property of class is described in a slot (Jain & Singh, 2013). Ontologies are commonly associated with taxonomic classes hierarchies, and but they are not limited to hierarchies, definition and relations.

Ontology is gaining a more important role in knowledge management, currently they are being used as Semantic Web standard knowledge representation tool. Ontology allows users to connect with among each other using a shared understanding for a domain. This shared understanding helps in understanding concepts of the domain as well as helps the machines to interpret the definitions of concepts in the domains as well as the relations between them (Ou, Orasan, Mekhaldi, & Hasler, 2008). To access ontology shaped structured data a query languages are required, SPARQL is one the formal query languages.

2.1.7 SPARQL

One of the ontologies query language is called SPARQL it is an acronym of Protocol and RDF Query Language, it is pronounced Sparkle, the SPARQL is able to retrieve and manipulate data stored in the Resource Description Framework format (El-Radie & Alagha, 2015). Some similarities between common SQL and SPARQL exist like Select, Where and other commands. Due to SPARQL purpose of querying ontologies some other words has been introduced like OPTIONAL, FILTER and others (El-Radie & Alagha, 2015).

RDF use is made possible by the flexibility and power of SPARQL, SPARQL with all of its advantages over traditional databases is powerful, flexible, and it allows the use of RDF. In order to make the SPARQL construction easier several methods have been suggested including assisted query construction (McCarthy, Vandervalk, & Wilkinson, 2012). SPARQL is built upon the concept of a triple pattern, this pattern is expressed as subject, a predicate and an object, a triple pattern sentence is closed by a dot. The SPARQL triple pattern can include variables: or the entire subject, the predicate, and the object values in the triple pattern can be a variable (Mitchell, 2013).

2.1.8 Normalized Discount Cumulative Gain

Information Retrieval is the process of retrieving information from the internet, that information are acquired by a user query sent over the web to a search service, in our research we design a search service which responds to the mobile device module, retrieves information and send them back to the user. In order to measure the Performance and correctness of our developed search service, we need to assess how well a system meets the information needs of its users. In order to do such assessment, traditional evaluation of an information retrieval system approaches has been designed for old search Boolean algorithms known as top-K retrieval and include precision and recall. Many other evaluation measures of performance of information retrieval were proposed, all of them considers a collection of documents to be searched and a search query. All the described measures are grounded of the document relevancy, every retrieved document to be either relevant or non-relevant to query as in (Järvelin & Kekäläinen, 2002).

Modern evaluation metrics are designed for ranking internet search results retrieval without any explicit rank cutoff, taking into account the relative order of the documents retrieved by the search engines and giving more weight to documents returned at higher ranks.

In order to evaluate the Performance and correctness of our developed search engine we evaluated the Discounted cumulative gain (DCG) to measure of ranking quality of

retrieved information, and it is the successor of other measure called CG, CG is a straightforward measure, which counts the appearance order of a document. We decided to use the DCG because it matches the method used in our developed search engine and serve our targeted goal. The DCG uses a graded relevance scale of documents in a search engine result set, it measures the usefulness, or gain, of a document based on its position in the result list. The gain is accumulated from the top of the result list to the bottom with the gain of each result discounted at lower ranks as in (Ahlqvist, 2015).

The DCG makes two assumptions:

- Highly relevant documents are more useful than marginally relevant document
- The lower the ranked position of a relevant document, the less useful it is for the user, since it is less likely to be examined two.

The DCG Uses graded relevance as a measure of usefulness, or gain, from examining a document, Gain is accumulated starting at the top of the ranking and may be reduced, or discounted, at lower ranks. The typical discount is 1/log (rank) with base 2, the discount at rank 4 is 1/2, and at rank 8 it is 1/3.

Assuming the relevance judgments are in a scale of [0,r]? r>2,

Using the Cumulative Gain (CG) at rank n

Let the ratings of the n documents be r1, r2, …rn (in ranked order)

CG = r1+r2+…rn

Using the Discounted Cumulative Gain (DCG) at rank n

DCG = r1 + r2/log22 + r3/log23 + … rn/log2n

DCG is the total gain accumulated at a particular rank p:

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2 i} \quad (2.1)$$

Also it can be calculated using other formula which emphasis on retrieving highly relevant documents

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{log(1 + i)} \quad (2.2)$$

In the following we show in example how DCG is calculated for a sample query. Table 4.4 illustrates the expert's rank given to the results of the query:"المصلى المرواني" and the GPS location coordinates latitude = 31.7722 and longitude = 35.228901 in Jerusalem, Palestine. The first column to the left shows the DBpedia URIs.The second column contains the webpage Label. The third column contains a serial, this serial is the order of the query returned results from our system as it returned them. The forth column contains the human expert's rank of the results. The fifth column contains the sorted human expert's ranks, the sixth column contains the $\log_{10} Serial$, the seventh column contains the multiplying of the fourth column (ExRange) with the sixth column (Log(serial)) and the eighth column contains the multiplying of the fifth column (SExRange) with the sixth column (Log(serial)) , to complete the evaluation we make sum for the seventh column(DCG_{10}) and make sum for the eighth column ($IDCG_{10}$) and divide the seventh to eighth column and multiply the result with 100% to get handed present which equal 93.8% with standard deviation 0.337.

Table (2.1): illustrate the evaluation metrics for the query results using normalized discount cumulative gain

Subject	Label	Serial	ExRange	SExRange	log_2^i	DCG_{10}	$IDCG_{10}$
http://DBpedia.org/resource/Solomon's_Stables	المصلى المرواني@ar	1	5	5	0	5	5
http://DBpedia.org/resource/Muristan	مورستان@ar	2	3	4	1	3	4
http://DBpedia.org/resource/Southern_Wall	الجدار الجنوبي (القدس)@ar	3	4	4	1.584963	2.523719	2.523719
http://DBpedia.org/resource/Jerusalem	القدس@ar	4	3	4	2	1.5	2
http://DBpedia.org/resource/Orient_House	بيت الشرق@ar	5	2	3	2.321928	0.861353	1.29203
http://DBpedia.org/resource/Mount_Zion	جبل صهيون@ar	6	1	3	2.584963	0.386853	1.160558
http://DBpedia.org/resource/Dung_Gate	باب المغاربة@ar	7	4	3	2.807355	1.424829	1.068622

http://DBpedia.org/resource/Gihon_Spring	نبع أم الدرج@ar	8	1	2	3	0.333333	0.666667
http://DBpedia.org/resource/Temple_Mount	الحرم القدسي الشريف@ar	9	4	1	3.169925	1.26186	0.315465
http://DBpedia.org/resource/Dormition_Abbey	كنيسة رقاد السيدة العذراء@ar	10	3	1	3.321928	0.90309	0.30103
		Summation				17.19504	18.32809
		Total percent				93.8179	

2.1.9 Mean Average Precision (MAP)

In order to further verify the results we used a second metric, it is called Mean Average Precision (MAP). These metric points out the average of the precision value obtained for the set of top retrieved existing k documents. Then precision obtained value is averaged over information needs.

MAP is calculated by using the following Equation:

$$MAP = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{Q_j}\sum_{i=1}^{Q_j}P(doc_i) \quad (2.3)$$

Where, N is number of queries, Q_j is number of relevant documents for query j and $P(doc_i)$ is precision at ith relevant document.

Ex., we calculated the average precision for each query and calculate the average of these averages.

To clarify the mean average precision, assume that we have two queries as the following:

16

Query 1		
Rank	Relev.	$P(doc_i)$
1	X	1.0
2		
3	X	0.67
4		
5		
6	X	0.50
7		
8		
9		
10	X	0.40
11		
12		
13		
14		
15		
16		
17		
18		
19		
20	X	0.25
AVG:		0.564

Query 2		
Rank	Relev.	$P(doc_i)$
1	X	1.0
2		
3	X	0.67
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15	X	0.2
AVG:		0.623

$$MAP = \frac{0.564 + 0.623}{2} = 0.594$$

This value indicates that system is 59% accurate is in retrieving relevant concepts.

This measure is applied on our dataset to calculate accuracy of relevant concepts. Then the mean average precision is calculated for 100 queries. Recall that results obtained for each query were rated by a human subject on a scale from 0 to 5. For the MAP measure, we assumed that a result is relevant if it is rated 3 or above. This assumption was based on similar studies (Agichtein, Brill, & Dumais, 2006; Clarke et al., 2008) .

Note that both nDCG and MAP are commonly used to evaluate recommendation systems and search engines. nDCG is mainly a measure of ranking quality, and uses a graded relevance scale of documents, e.g. a relevance scale from 0 to 5. MAP is a measure of quality as it measures how relevant the retrieved results are. Unlike nDCG, MAP uses a binary relevance scale, e.g. relevant or not relevant.

Table 4.5 illustrate the Expert rank to the results of the query with the user search words "المصلى المرواني" and the GPS location coordinates latitude = 31.7722 and longitude = 35.228901 in Jerusalem, Palestine the first column is the DBpedia webpage subject the second column contains the webpage Label, the third column contains a serial the forth column contains the human expert rank for the results, the fifth column contains the relevant result , the sixth column contains the $P(doc_i)$, the seventh column contains the mathematical calculation from $P(doc_i)$ column then at the last of the table make

summation for the Result column was equal 6.08095 divide it on relevant results count 7 the result will be 0.8687 and multiply it by 100 get handed present which equal 86.87%.

Table (2.2): illustrate the evaluation metrics for the query results using MAP metric.

Subject	Label	Serial	ExRange	Relev	$P(doc_i)$	Result
http://DBpedia.org/resource/Solomon's_Stables	المصلى المرواني@ar	1	5	X	1/1	1
http://DBpedia.org/resource/Muristan	مورستان@ar	2	3	X	2/2	1
http://DBpedia.org/resource/Southern_Wall	الجدار الجنوبي (القدس)@ar	3	4	X	3/3	1
http://DBpedia.org/resource/Jerusalem	القدس@ar	4	3	X	4/4	1
http://DBpedia.org/resource/Orient_House	بيت الشرق@ar	5	2			
http://DBpedia.org/resource/Mount_Zion	جبل صهيون@ar	6	1			
http://DBpedia.org/resource/Dung_Gate	باب المغاربة@ar	7	4	X	5/7	0.7143
http://DBpedia.org/resource/Gihon_Spring	نبع أم الدرج@ar	8	1			
http://DBpedia.org/resource/Temple_Mount	الحرم القدسي الشريف@ar	9	4	X	6/9	0.6667
http://DBpedia.org/resource/Dormition_Abbey	كنيسة رقاد السيدة العذراء@ar	10	3	X	7/10	0.7
	Summation					6.081
	MAP Result					0.8687

2.2 Related Works

In order to achieve the work in this paper other works have been investigated. This section contains some of the investigated works, works are used to support and justify my approach. This section includes a summary of the main related recommendation systems that use the Context-Aware, Mobile recommendation Systems, DBpedia and/or Wikipedia as a source of information.

2.2.1 Context-based Recommendation Systems

In (Adomavicius, Sankaranarayanan, Sen, & Tuzhilin, 2005) the researchers claims in order to increase the quality of produced recommendation it is vital to infuse the contextual information in the recommendation systems. In his paper (Adomavicius et al., 2005) the researcher described a multidimensional approach to recommendations, where the user profile matrix by adding new dimensions elements including, time, place. In general, as mentioned in (Adomavicius & Tuzhilin, 2011) context-aware recommendation system can be in one of three forms, these forms can be named depending on the stage of using the context in producing a recommendation, and the three forms are contextual pre-filtering, Contextual post-filtering and Contextual modeling. The three forms compared in (Panniello, Tuzhilin, Gorgoglione, Palmisano, & Pedone, 2009) this comparison is based on some experiments were the researchers suggest simple and effective method for using two methods in a recommendation system.

The researchers in this paper (Ostuni et al., 2012) tries to produce a movie recommendation system called Cinemappy, a content-based location aware recommendation system. This system is able to process contextual movie recommendation and refines the contents used spatial filter both current and temporal user location. The DBpedia is the source of information for this content-based recommendation system.

In (J.-D. Zhang & Chow, 2016), the researcher makes a time-aware recommendation using a the Temporal Influence in his proposed method, this method combines use based and location-based correlations in recommend time to visit a location.

In (Masthoff, Mobasher, Desmarais, & Nkambou, 2012), the researcher assumes that similarity among contextual situations should produce similar recommendation lists. This method produces a context aware recommendation by learning context similarities, time information is one of the context.

In (W. Zhang & Wang, 2015), the researcher proposed a new model that takes into consideration the friendship among users, it combines the users relations the current user location and the time of information. This proposed model is called location and time aware social collaborative retrieval model (LTSCR).

In (Muntean, Nardini, Silvestri, & Baraglia, 2015), the researcher tries to suggest the user next point of interest based on his history of preferences. those preferences are collected using a supervised learning techniques. a 68 features are composed in order to describe the user feature sets. those 68 features include time-based features to model how users spend their available time.

2.2.2 Mobile Recommendation Systems

As mobile smart devices are becoming more popular and they can provide several services to clients. Many developers are working on developing mobile device based recommendation system. Mobile devices recommendation systems can offer personalized context-sensitive recommendations. The context of the user such as the location, time and the surrounding environment, can be exploit to give more personalized content. However, mobile-based recommendation systems face deferent challenges: First, they are considered more complex due to the necessity of working with heterogenic, and noisy data. Second, they have to consider spatial and temporal circumstances and build some auto correlations (Ge et al., 2010).

Smart mobile devices do have a GPS system which can greatly benefit in providing rout recommendation. For example, it can recommend the most suitable driving routes for taxi drivers in the city. (Ge et al., 2010) Such systems take input from the device's GPS and trace the taxi drivers routes in the city including the longitude, latitude, timestamps and status of operation (occupied/busy).

MobyRek (Ricci & Nguyen, 2007) is a mobile device recommendation system designed to help travelling users in searching for travel products. this recommendation system is based on asking and answering a set of questions, the recommendation is made in cooperation with other Web based recommendation system called NutKing. Nutking is a system built to help users plan and build their travel plans. Nutking uses the Content based recommendation approach. The on-tour support is used when a mobile device user traveler with or without a pre-travel plan is on the way to or at the selected destination. the Nutking is designed to meet two general requirements. First, the product recommendations are relevant to the user's specific preferences. Second, the user-system interaction is simple, requiring minimal time to obtain a useful recommendation.

In (Bouneffouf, 2013), the researcher developed a dynamic exploration/ exploitation strategy to improve with the context aware mobile recommendation systems . the developed method can automatically learn the optimal tradeoff objective and adaptively balance the two aspects of exploration/ exploitation. the used approach consists of optimizing a utility function of the clicked and the non-clicked documents, which are already recommended. the used approach uses the content filtering based on the user context.

In (V. W. Zheng et al., 2010), the researcher tries to build a Collaborative Location and Activity Recommendations with GPS History Data, this recommendation system will try to make recommendations to mobile users based on their GPS Location and recommend them with activities and other sites to visit based on other users experiences. The system mines knowledge including location features and activity-activity correlations from the geographical databases and the Web. the recommendation system uses the built knowledge base to recommend to the users where they can visit if they want to perform some specific activities and what they can do if they visit some specific places.

In (Vagliano et al., 2016), the researcher Proposed a A Dynamic Recommendation Algorithm system called ReDyAL, the Proposed ReDyAl is a hybrid algorithm, it discovers resources using both the traversal and hierarchical approach dynamically. this Algorithm is application domain independent and can be easily adapted to other dataset in the Web of Data, in this work it has been applied on the DBpedia. data set reduction in a specific domain is not required in the ReDyAl.

In (Jung & Chung, 2016), preventive management is replacing medical diagnosis and treatment leading to taking over the conventional health management, context-aware modeling is getting growing attention around the world. Youth obesity problem is growing and causing serious problems in most of the modern diseases. this student proposes a based on obese youth dietary nutrition recommendations. the proposed system goes beyond static dietary nutritional data to reach individualized diet menus for them by utilizing knowledge-based context data through a collaborative filtering method. the proposed system utilizes not only the basic information of the user, but forms similarity clustering and correlation, Unlike the conventional uniformed dietary nutrition recommendations for obesity management, the proposed method is capable of providing personalized recommendation, and provide the user with personalized recipes and menus on their mobile phones and time anywhere.

2.2.3 Mobile Recommendation Systems of Locations
The vast and rapid development of mobile applications that provides a great amount of data of all types (images, texts, sounds, videos, etc. those information can be used in Mobile Context-aware recommendation Systems (MCRS) which can suggest suitable information to the user this suggestion is based on her/his location, circumstances and interests. recommendation systems must produce and suggest individual information to each user based on the information collected by his mobile phone, these suggestion are carefully chosen from a large number of alternatives. content-based filtering or collaborative are used in the recommendation systems (Bouneffouf et al., 2012).

Some research works tried to take the user's situation into consideration recommendation. In (Bellotti et al., 2008; Bouidghaghen, Tamine-Lechani, & Boughanem, 2009; Panayiotou & Samaras, 2006) the authors proposed an approach which is consisted of building a dynamic situation and user profile based on time and user's experience. The user's preferences and interests saved in the user profile are measured according to the circumcises (time, location) and user actions and behaviors. These information are used to model the alterations of user's preferences according to his time-based situation in different points of time, these times can be workday or vacations, the measured association for the concepts in the user profile is made for every new user experience. User activity are combined with the user's profile then used together to choose and recommend relevant content to the user.

A deferent approach has been tested in (Ramaswamy et al., 2009) this approach Process MRCS operation using three dimensions of context, these dimension complete each other in order to get the best recommendation. The MCRS starts by analyzing clients' information such as address books, used to estimate the level of social relations among users. Then it combines social relation with the location and time (spatiotemporal) dimension, finally this information are checked against user's history in order to improve the quality of the recommendations.

The authors In (de Spindler et al., 2006) present a user based collaborative filtering technique. Each user's stores his explicit ratings in his mobile device; Other ratings are received from other users. Only users in spatiotemporal proximity are able to exchange ratings. This approach provides a natural filtering based on social contexts.

As mobile application are most attractive for Context-aware recommendation application, because the mobile devices give the users access to enormous amount of information in a universal way. Many issues, scenarios and opportunity are mentioned in (Bouneffouf, 2013) many of those issues are about travel and tourism. The author describes many of the major used techniques and some specific computational models, which have been proposed for mobile recommendation systems. COPASS is a context-aware mobile tourist application; it is suggested in (Bizer et al., 2009), COMPASS models the current user requests while considering his needs, profile and context information, later compass performs a selection of potentially interesting close landmarks and objects. The information collected by compass are updated and changed when the user changes his location or target. The authors of (Howland) Present ReRex, which is another proposed context-aware mobile recommendation system, ReRex suggests the Points Of Interests POI based on web-Based survey application, the users need to make a contextual condition and then assess a POI. This assessed POI is used in

the application to let the user chose a contextual factor and browse a related context-aware recommendation.

In (De Pessemier, Dooms, & Martens, 2014), the researcher proposed a context recognition framework, this framework recognizes the user current context and activity using the mobile device accelerometer and sensor data. the framework monitors and process the collected data in order to recognize basic activities and/or context change, later those activities are analyzed in order to recognize the overall context of the user. this framework proved to be effective and battery efficient, it provides to context needed for a recommendation system without using the GPS.

In (Rawat & Kankanhalli, 2017), the researcher proposed a viewpoint recommendation system ClickSmart, which can help the user in shooting good photographs at tourist locations. Clicksmart can provide instant viewpoint recommendation based on the user's camera preview, time and user geolocation. The clickSmart uses the publically available geotagged images with its metadata for learning a recommendation model. furthermore ClickSmart observe contextual information such as time and weather conditions to improve the recommendation system with the associated context.

2.2.4 Discussion

Our initiative is distinguished over the previous efforts in the following points:

1. Existing recommendation systems often relied data repositories and proprietary geo-location services, those services do not cover areas of the world, including the middle east specially the Gaza strip. In this work we use an open-source data DBpedia, which is open and can accommodate information About every place in the world.
2. In this word the LOD is used in order enquire the DBpedia, to retrieve content and geo-locations.
3. Existing recommendation systems often rely on existing structures and categorization of geographical places to identify the user's interests (i.e. categorizing places into restaurants, museums, theatres, etc). This work aims to identify the categories of places automatically by analyzing the DBpedia resources.

Chapter 3
Methodology

Chapter 3   Methodology

3.1 Introduction

The widespread use of smart phone devices makes them the most targeted devices for application development. A considerable number of mobile applications utilize the unique features offered by mobile devices such as proximity and location. The use of these features allowed the development of applications that adapt content with the context of the user.

In this chapter, we propose a recommendation system that exploits DBpedia and location based services to recommend locations relevant to a user query. In the first section, we explain our design principles, and the system architecture is explained in detail, focusing on the main steps of the system search for related information, including the: search keyword extraction, user localization, DBpedia based location finder, Query preprocessing, and the ranking algorithm.

3.2 User Scenario

To illustrate how the system works, we give the following scenario: Assume that a user is located in Gaza city, and he/she uses the mobile device to search for tourist areas nearby. Once the user opens the application, a map will be shown on the mobile screen and a marker on his location on the map will appear as shown in Fig. 3.1. From the top bar, the user can search for places of interest by inputting keywords. If the user types the keywords: "tourist places" and launches the search service, the user's position and keywords will be sent through a web service to the server side of the system. The user's request is processed on the server, and results is returned back and displayed on the screen as multiple markers as shown in Fig. 3.2. These markers refer to tourist locations in the Gaza city that are close to the user's location. Clicking on any marker will open the corresponding Wikipedia page that presents details information about the location.

The user will know about the importance of the location by a scale of stars displayed next to each location title on the displayed map (see Figure 3.2). The more stars the location have, the more important and relevant it is to the user.

It is important to notice that the resulting locations are not defined on Google maps, or on any other mapping service. In fact, most mapping services do not provide detailed information about locations in Gaza. Nevertheless, our system could retrieve several locations relevant to the user query even if these locations are not defined by the widely used mapping services. These locations were identified and retrieved by referring to

Open Linked Data (LOD) which our system uses as an alternative to mapping services for uncovered areas.

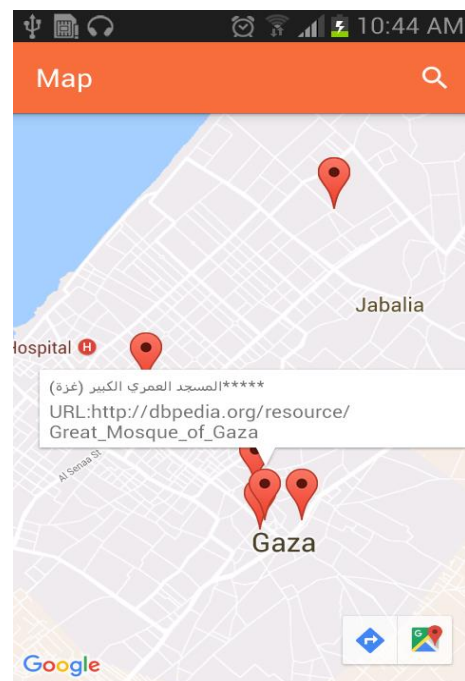**Figure (3.1):** illustrate the Clint side interface

Figure (3.2): illustrate the Clint side interface with the location around the user

3.3 Design Principles

After illustrating how the system works, we discuss the design principles of the system:

- Exploit LOD as an alternative to mapping services for unidentified locations: Many geographical areas are not widely covered by common mapping services. For example, the map of the Gaza strip does not present detailed information about locations in Gaza. It also does not provide a classification of these locations (e.g. tourist areas, schools, universities, etc.). Therefore, our main motivation for designing the system was to seek an alternative solution to common mapping services for areas not covered in detail. LOD resources, such as DBpedia, has almost 50 resources that refer to locations in Gaza. The proposed system will exploit these resources to identify locations relevant to the user query on the fly, and present them to the user on the map.
- It is important to notice that the LOD based approach does not aim to replace mapping services. In fact, mapping services still have wider coverage. However, we think that the combination of both LOD and mapping services will provide the best coverage and results.
- Balance between proximity and relevancy: The system should make balance between relevant locations and the distance to the user's location. Some locations may be very relevant but they are distant from the user's current position. On the other hand, other locations may be close to the user's position but do not match to his/her interests to a large extent. The underlying recommendation algorithm should balance between proximity and relevance when ranking the recommendation results.
- Provide details about retrieved locations: we also aimed not only to recommend locations, but also to provide details on these locations so that the user can gain an overview about these locations at the glance. Therefore, our system will associate each marker with a link to the corresponding Wikipedia page.

This research has been carried out by developing a mobile-based recommendation system that takes keywords from the user and the location from the mobile GPS as inputs.   It then returns results related to both location and keywords as output. The system implicitly refers to DBpedia to infer information that links the location with the input keywords.

3.4 The Architecture of the System

The system architecture is illustrated in Figure (3.3). It consists of two main components: the client side and the server side:
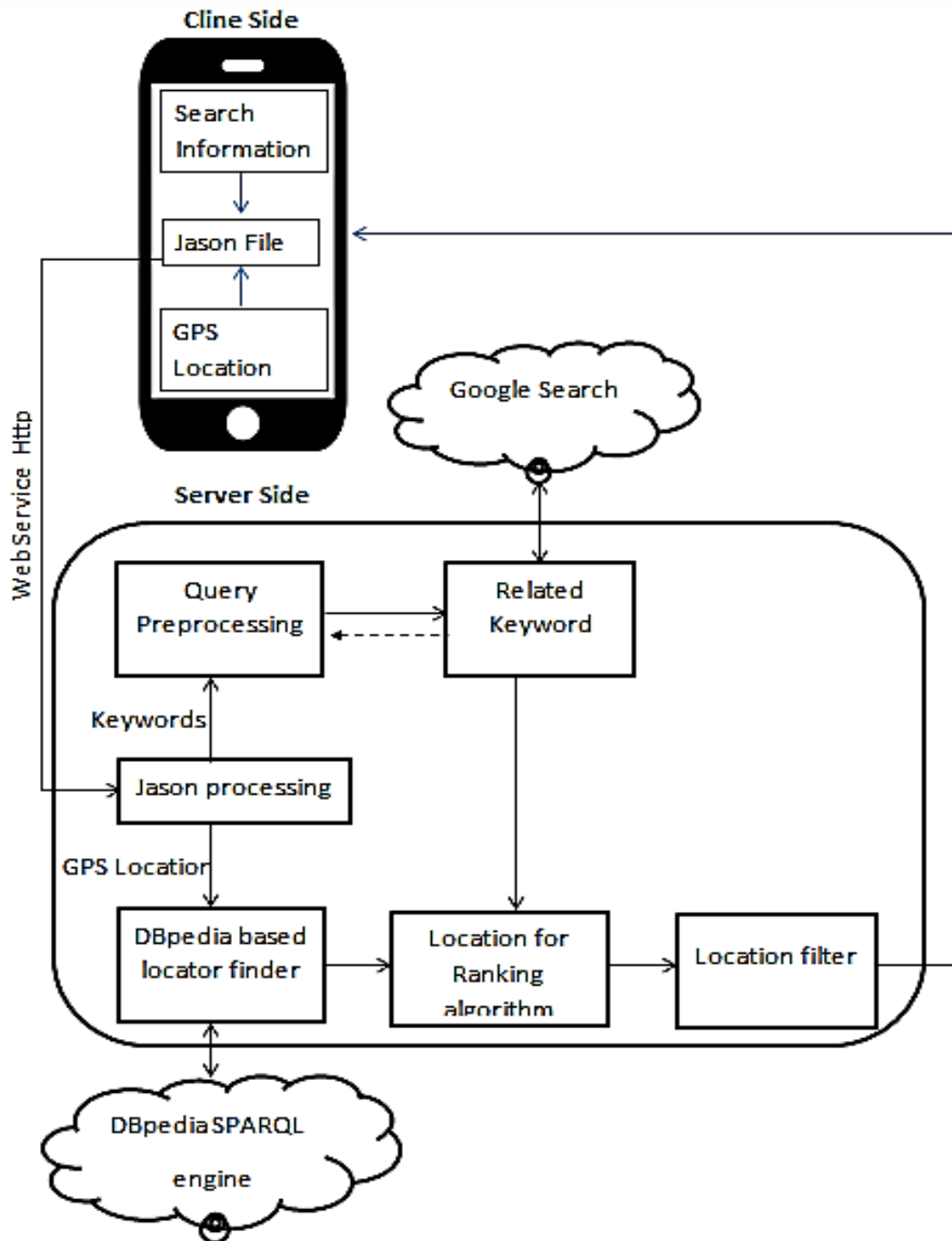

Figure (3.3): illustrate the system architecture

3.5 The Client Side

The client side is located on the mobile device. It is a lightweight component and has two jobs: the first is to get the location of the user from the mobile GPS system and the search keywords which the user inputs using the application interface. The user location and the search keywords are sent to the server side through a Restful web service. The client side creates a JSON file which contains the user location GPS latitude and longitude and the searched keywords as illustrated in JSON snippet shown in Figure 3.4. In the example shown in Figure 3.1, the user inputted the keywords: سوق الذهب. And sends it to the server side through a web service over the Internet.

{"keyword" : "سوق الذهب " , "lat":31.504203,"long" : 34.464467}

Figure (3.4): illustrate the JSON file structure sends from the client to the server

the second job of the client side is to receive the results from the server side as a JSON file. The received results file will contain the places information like place title, GPS coordination, abstract, and the URL. Received results in JSON will be visualized so that it can be presented to the user as markers on the map as shown in Fig 3.2, the user can click the location's marker to view the location's title or visit the locations' pages on DBpedia.

The client side was created as a lightweight module in order to make it easy to develop and operate on the any smart phone operating system like android, IOS or windows mobile. The current prototype of the client side was developed for Android OS.

3.6 The Server Side

The server side component resides on the server, and Its main job is to receive the keywords and user's location from the client side. It then performs whole processing and sends the results back to the user. This server side component consists of six main modules as the following:

3.6.1 JSON Processing Module

This module receives the JSON file containing search keywords and user's location from the client side, and then processes it. The module performs two types of processing: The first Processing includes extracting the search keywords and sending them to the Query Preprocessing module. The second processing of the JSON Processing module is to extract the user GPS location (latitude, longitude) and sends them to the DBpedia based location finder module in order get the places around the user location. These two processes are explained in detail in the following subsections:

3.6.2 DBpedia Based Locator Module

The DBpedia based locator module aims to identify the user's location based on the information from DBpedia. As mention earlier, our system relies on LOD to identify locations that may be missing on conventional mapping services. The DBpedia is an open source structured Data; it is built on top of the Wikipedia. It contains a wide variety of information and data. A huge amount of these information are about places and Geographic locations whose information are organized as pages, the Geographic location pages have a properties for GPS coordinates (latitude, longitude, sometimes altitude) with the names:(geo:geometry, geo:lat, geo:long) as illustrated in Figure. (3.5).

Our hypothesis is that the user's location obtained from the GPS system can be identified by searching DBpedia for matching GPS coordinates. Afterwards, information about the user's location can be retrieved from the corresponding DBpedia page and provided to the user.

dbpedia.org/page/Gold_Market

DBpedia   ◉ Browse using ▾   📄 Formats ▾   ⬈ Faceted Browser   ⬈ Sparql Endpoint

About: (سوق الذهب (غزة

An Entity of Type : building, from Named Graph : http://dbpedia.org, within Data Space : dbpedia.org

سوق الذهب في غزة (ويعرف أيضا باسم سوق القيسارية) هو ممر ضيق مغطى يقع في الحي القديم في مدينة غزة , وهو مركز لتداول وشراء الذهب , وموقع لصرف العملات الأجنبية . يقع السوق إلى جانب الحافة الجنوبية من الجامع الكبير في غزة(مسجد العمري) . بجانب شارع عمر المختارالرئيسي . السوق مغطى بسقف مقبب وعلى جانبيه محلات الذهب ذات شكل مميز وصغير وهي عبارة عن غرف صغيرة ملفوفة الشكل .

Property	Value
geo:geometry	• POINT(34.463890075684 31.50305557251)
geo:lat	• 31.503056 (xsd:float)
geo:long	• 34.463890 (xsd:float)

Figure (3.5): illustrate the DBpedia location page with prosperities about GPS (latitude, longitude)

When we search DBpedia for geographical locations in Gaza, we found about fifty DBpedia pages referring to geographical locations in the Gaza city in Palestine, and fifty DBpedia pages referring to geographical locations in Jerusalem city. Examples of these locations are included in Table (3.1). We could also find about one hundred and fifty DBpedia pages referring to geographical locations in Salah Salem street in Cairo, Egypt. All these DBpedia pages have the geographical properties: geo:geometry, geo:lat, and geo:long. Although this number of DBpedia pages is relatively small to be used as an alternative to mapping service, it can be used as a proof of concept, and to illustrate how DBpedia can be used to answer user queries when traditional mapping services fail to do so. It is worth noticing here that we do not aim to replace the mapping services, but to

extend them by exploiting geographical information defined in DBpedia. One should notice that DBpedia is rapidly growing, and it is likely that the geographical locations on DBpedia will considerably increase in the near future.

Another potential advantage of our system is that it tries to semantically find locations related to the user query. It does this by exploiting the contents of DBpedia pages and by expanding the input user query. In contrast, traditional mapping services often depend on predefined classifications of locations to identify related locations. Therefore, these mapping services will not be able to find related locations if predefined classifications do not exist.

Given the user's location (i.e. latitude and longitude), a SPARQL query will be used to query DBpedia for all resources that refer to locations close to the user's position. The properties geo:lat and geo:long in DBpedia resources store the location's coordinates, which will be compared with the user's location. To make the query tolerant, we search for all locations within a radius of 2 kilometers. Later on, retrieved locations will be filtered and ranked to keep only the most relevant and close ones.

 In order to query DBpedia and get the places around, the user's location will be obtained from the GPS, and will be used to build a SPARQL query that retrieves from DBpedia information as a result about the around places DBpedia pages. Assume the GPS location latitude = 31.5131 and longitude = 34.4405, which refers to the Islamic University of Gaza, and we need the places around it approximately to two kilometers around, the SPARQL query in Figure (3.6) will be constructed.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>PREFIX

geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>PREFIX

dbo: <http://dbpedia.org/ontology/>PREFIX

rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX type: <http://dbpedia.org/class/yago/>

PREFIX prop: <http://dbpedia.org/property/>

PREFIX vcard: <http://www.w3.org/2006/vcard/ns#>

PREFIX vc:    <http://www.w3.org/2001/vcard-rdf/3.0#>

SELECT DISTINCT ?subject ?label ?lat ?long ?abstract

        WHERE {

                ?subject geo:lat ?lat.

                ?subject geo:long ?long.

                ?subject rdfs:label ?label.

                ?subject  <http://dbpedia.org/ontology/abstract> ?abstract

        FILTER (?lat - 31.5131 <= 0.05 && 31.5131 - ?lat <= 0.015 &&

                ?long - 34.4405 <= 0.015 && 34.4405 - ?long <= 0.05 &&

                lang(?label) = "ar" &&

                lang(?abstract) = "ar"). }
```

Figure (3.6): illustrates the SPARQL query to get the around locations to the Islamic
University of Gaza.

The results of the above SPARQL query are illustrated in the Table (3.1). The table
shows columns for subject (URI to DBpedia resource), label as mentioned in DBpedia,
latitude, and longitude.

31

Table (3.1): illustrate the result received from the DBpedia using SPARQL query

Subject	Label	latitude	Longitude
http://DBpedia.org/resource/Sayed_al-Hashim_Mosque	مسجد السيد هاشم@ar	31.5081	34.4633
http://DBpedia.org/resource/Sheikh_Ijlin	الشيخ عجلين@ar	31.5134	34.4655
http://DBpedia.org/resource/Al-Azhar_University_–_Gaza	جامعة الأزهر بغزة@ar	31.515	34.4367
http://DBpedia.org/resource/Qasr_al-Basha	قصر الباشا@ar	31.5044	34.466
http://DBpedia.org/resource/Great_Mosque_of_Gaza	المسجد العمري الكبير (غزة)@ar	31.5042	34.4645
http://DBpedia.org/resource/Great_Mosque_of_Gaza	المسجد العمري الكبير (غزة)@ar	31.5042	34.4645
http://DBpedia.org/resource/Ibn_Uthman_Mosque	جامع ابن عثمان@ar	31.5042	34.4697
http://DBpedia.org/resource/Port_of_Gaza	ميناء غزة@ar	31.5258	34.4306
http://DBpedia.org/resource/Gold_Market	سوق الذهب (غزة)@ar	31.5031	34.4639
http://DBpedia.org/resource/Church_of_Saint_Porphyrius	كنيسة القديس برفيريوس@ar	31.504	34.462
http://DBpedia.org/resource/Karni_crossing	معبر المنطار@ar	31.4747	34.4736
http://DBpedia.org/resource/Rimal	الرمال (غزة)@ar	31.5308	34.4558

Subject	Label	latitude	Longitude
http://DBpedia.org/resource/Islamic_University_of_Gaza	الجامعة الإسلامية (غزة)@ar	31.5131	34.4405
http://DBpedia.org/resource/Sheikh_Radwan	الشيخ رضوان (غزة)@ar	31.536	34.4658
http://DBpedia.org/resource/Muhammad_al-Durrah_incident	مقتل محمد الدرة@ar	31.4651	34.4267
http://DBpedia.org/resource/Al-Shifa_Hospital	مستشفى الشفاء@ar	31.5241	34.4442

The SPARQL query sent from the server side to the DBpedia returns all the locations around the user's location from the DBpedia.

A relation between the user's interest and the query result must be made, therefore another module is developed to filter the returned SPARQL query results based on the user search keywords. This module should keep locations that the user is interested in, and filter out the rest. This will make the system more practical and friendly to the user.

3.6.3 Query Preprocessing Module
Our system does not depend only on getting the around places from the DBpedia using the SPARQL query, but there is an important part of the system which analyses the user query to find if the place has a relationship with the user search words. Before finding this relationship, this module performs basic NLP techniques on the search query entered by the user. This includes three stages: the first stage is tokenization: Tokenization is the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens. Tokens can be individual words, phrases or even whole sentences. In the process of tokenization, some characters like punctuation marks are discarded. The tokens become the input for another process like parsing and text mining. Tokenization is used in computer science, where it plays a large part in the process of lexical analysis.

The second stage is light stemming: This method is used to find out the stem of a word. For example, the words user, users, used, using all can be stemmed to the word "USE". The purpose of this method is to remove various suffixes, to reduce number of words, to

have exactly matching stems, to save memory space and time. The stemming process is done using various algorithms. Most popularly used algorithm is "M.F. Porters Algorithm.

The third stage is stop word removal: Most frequently used words in English are useless in Text mining. Such words are called Stop words. Stop words are language specific functional words, which carry no information. It may be of the following types such as pronouns, prepositions, conjunctions.

3.6.4 Query Expansion Module

The Query Processing Module results in the preprocessed keywords from the user query. The preprocessed keywords are then inputted to the Query Expansion Module which aims to expand the user query by retrieving more keywords related to the user's keywords. The expansion of the user query with additional related keywords will help to better identify DBpedia resources that potentially match with the user interests. In fact, there are many ways to make this expansion: one way is by using WordNet. WordNet it is a dictionary that gives synonyms for words, and has been widely used as background knowledge for many information-based system (Rodríguez et al., 2008). Although there exists an Arabic version of WordNet, we could not find any library that supports rapid access and search in its content. Therefore, we decided to use the Google search service to achieve the desired expansion as the following:

The user's keywords are sent to the Google search service. The first 20 snippets from the search results are extracted and preprocessed by removing stopwords, tokenization and light stemming. For example, if the user inputs search query (سوق الذهب في غزة) after make preprocessing it will be (سوق الذهب غزة) then get the word snippet from the Google search service the result well be as illustrated in the Table (3.2) every returned word from the Google search service is associated with its rank in word snippet. The rank is the order of the search result containing the word in all results. Lower ranks denote more important results because they come first. The rank is important because the importance of the word depends on its rank in the search snippet. Words in the first rank will be more important than words in the second rank and so on. The word rank will be taken into account while filtering the locations in a later stage.

Table(3.2): illustrate the snippet word returned from Google search service, the first column contains the word and the second column contains the word rank as in Google search service.

Word	Rank	Word	Rank	Word	Rank	Word	Rank	Word	Rank
كانون	1	أسعار	4	أبو	8	للذهب،	12	نوعاً	15
يناير	1	فلسطين	4	بدر	8	كبير	12	المحللون	15
سوق	1	الأيام	4	تموز	9	مقارنة	12	و	15
الذهب	1	السابقة	4	يوليو	9	بمساحة	12	لو	15
غزة	1	السوق	4	سما	9	القطاع	12	الأول	16
حلوة	1	الفلسطيني	4	المقرر	9	الصغيرة	12	ديسمبر	16
يا	1	يتم	4	أن	9	تجوّل	12	حي	16
دنيا	1	متابعة	4	تجرى	9	المونيتور	12	الدرج	16
تقرير	1	أسعار	4	الجمعة	9	ميدانياً	12	بالبلدة	16
عمّان	1	اليوم	4	المقبلة	9	وسط	12	أهم	16
ويعرف	2	النصيرات	5	انتخابات	9	غزّة،	12	اسم	16
أيضا	2	جديد	5	جمعية	9	شباط	13	نظراً	16
القيسارية	2	متوفر	5	اصحاب	9	فبراير	13	لان	16

Keywords resulting from the query expansion process will be used to identify most relevant DBpedia resources. These keywords will be used in the Ranking algorithm module which is explained in the subsequent section.

3.6.5 Ranking Algorithm Module

The DBpedia locator module identifies DBpedia resources that refer to locations near the user's position within a predefined distance. However, not all of these locations match with the user's needs. Therefore, these locations should be filtered and ranked so that only most relevant locations are maintained and presented to the user on the map. The ranking algorithm is shown in Figure (3.7).
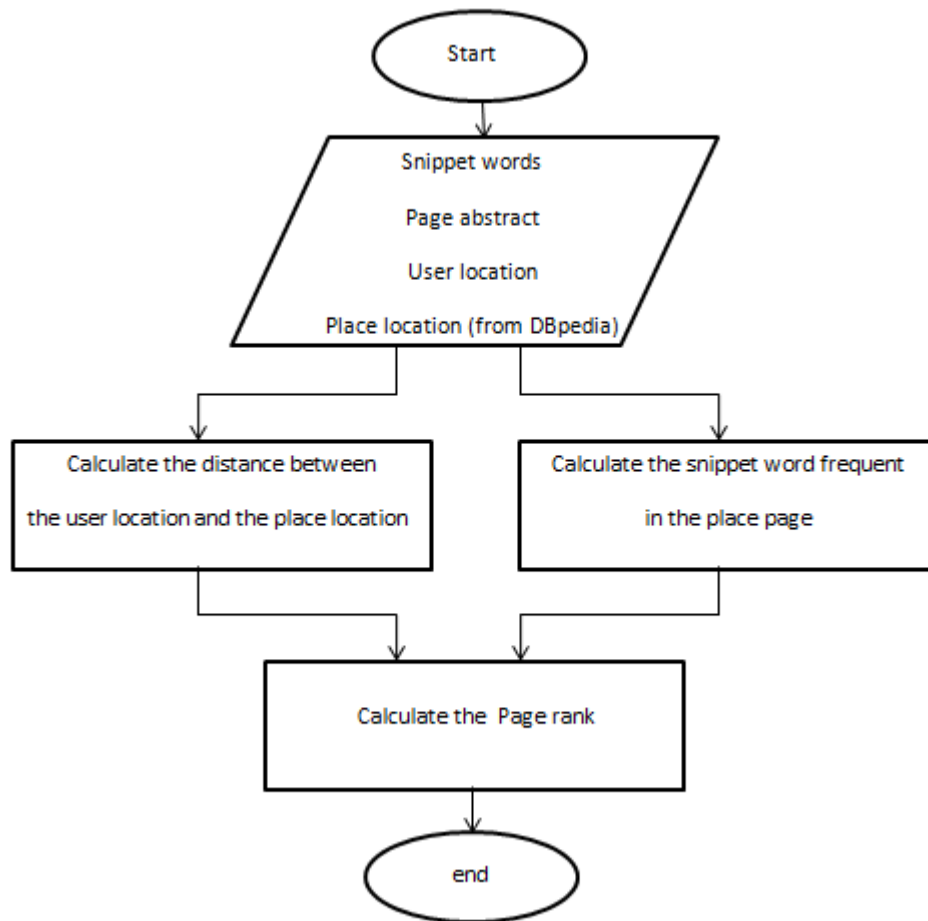
Figure (3.7): illustrates the results ranking process

The Query Expansion Module aims to identify keywords related to the user query. These additional keywords will help in filtering and ranking the retrieved DBpedia resources.

When ranking DBpedia resources that refer to locations, two factors should be considered: the proximity to the user's position, and the relevance to the user's input query. The proximity to the user's position is determined by calculating the distance between the user's coordinates and the latitude and longitude properties retrieved from DBpedia resources.

The relevance to the user's input query is determined by exploiting the keywords retrieved from the query expansion step. The input to the ranking algorithm includes: the keywords resulted from the query expansion process, the user's location, and the DBpedia based location and abstract are extracted from DBpedia resources. Abstracts of DBpedia resources are searched for words that match keywords resulted from the query

36

expansion step. The more the abstract contains matching words, the more relevance score will be assigned to the DBpedia resource. Therefore, the system will not choose location close to the use location but it has no relation with the user search query word, this module receives the user location and the DBpedia web pages' location and calculates the distance between the user location and every DBpedia web pages' location and normalize the distances to arrange from zero to one. The second input to this module is the Google search service word snippet and the DBpedia location web page abstract to count the frequency of the snippet words in the location abstract task is calculating the place DBpedia page according to the search keyword and its related word, which was, received from google word snippets using the mathematical equation (3.1)

$$Relevance\ score = \frac{1}{n} \left(frw_1 + \sum_{i=2}^{n} \frac{frw_i}{\log_2 rank_{wi}} \right)$$

<center>Mathematical Equation (3.1)</center>

Where:

W_i is the word resulted from the query expansion process.

$Fr(W_i)$ is the frequency of word W_i in the abstract of DBpedia resource

Score of W_i is the rank of W_i, which is the order of the search snippet that contains W_i. It is assumed that search snippets that come first are more related to the user's keywords. Therefore, the highest the score of W_i, the more relevance it is.

Where n is the number of the related words retrieved from Google snippets, $Fr(W_1)$ is the frequency of count number one ranked word in the DBpedia Page, the summation is the sum of the frequency on number 2 to n ranked word, and score W_i is the word order in the google snippet words, where the $\log_2$ became larger when the score larger than for example $\log_2 3 = 1.58$ larger than $\log_2 2 = 1$ that the important of the word become less than when the score become larger.

For example, the calculation of the mathematical equation illustrated in Table (3.3) in this table the column No. is serial the columns Word contains the snippet words the column rank contains the word rank in the snippet words as in google search service the column Count contains the frequent of the word in the abstract of the location DBpedia web page as in the equation (3.1).

<center>37</center>

Table (3.3): illustrate the frequent snippet word in the abstract of location DBpedia web page.

No.	Word	Rank	Count	No.	Word	Rank	Count
1	الخلافة	1	0	24	شكل	1	0
2	الإسلامية	1	0	25	إحدى	1	0
3	الراشدة	1	0	26	عَنْ	2	0
4	الأموية	1	0	27	رَسُولِ	2	0
5	العباسية	1	0	28	اللهِ	2	0
6	القرطبيَّة	1	0	29	صلى	2	0
7	المسجد	1	7	30	الله	2	0
8	أول	1	1	31	وسلم	2	0
9	مبنى	1	0	32	أنّهُ	2	0
10	تشهده	1	0	33	قَالَ	2	0
11	المدينة	1	0	34	قَدِ	2	0
12	المنورة	1	0	35	اجْتَمَعَ	2	0
13	العاصمة	1	0	36	فِي	2	0
14	الأولى	1	1	37	يَوْمِكُمْ	2	0
15	للدولة	1	0	38	هَذَا	2	0
16	مباشرة	1	0	39	عِيدَانِ	2	0
17	وصول	1	0	40	فَمَنْ	2	0
18	النبي	1	0	41	شَاءَ	2	0
19	محمد	1	0	42	أجْزَأَهُ	2	0
20	الصلاة	1	0	43	مِنَ	2	0

No.	Word	Rank	Count	No.	Word	Rank	Count
21	والسلام	1	0	44	الْجُمُعَةِ	2	0
22	مهاجرا	1	0	45	وَإِنَّا	2	0
23	مكة	1	0	46	مُجَمّعُونَ	2	0

The total summation for the column Count is nine and the total count of the snippet words is forty-seven then the final result from this location came from the equation (3.1) is 0.195652174. The results cannot expect the range of it so the system normalize the results between zero and one.

The distance between the user GPS location and the DBpedia GPS locations is calculated and normalized in order to fit into the prioritizing equation, the priority of the location is calculated by dividing the calculated distance on the max allowed distance, this gives us a value between 0 and 1, where 0 is the highest priority and 1 is the lowest. That location information will be obtained from the DBpedia, the relevance between the page and user interest is calculated based on the user search keywords. The distance priority and page relevance is calculated based on the following mathematical equation (3.2) in order to get the best page to the user, Finally the results are sent to the client side to be displayed.

$$Page\ rank = \propto * (1 - ND) + \beta * Relevance\ score$$
Mathematical Equation (3.2)

Where $\propto$ *and* β are fixed variables and $\propto + \beta = 1$, ND is the Normalized Distance between the user and the location, and the *Relevance score* is the result of Mathematical Equation(3.2) .

3.6.6 Results Builder Module

DBpedia resources are ranked based on Equation 3.2, which considers both the page relevance (Rank) and the proximity to the user's location (ND). These results should be presented in a way that can be easily perceived by the user. For each DBpedia resource, the Results Builder Module will retrieve the page title, abstract, GPS location (latitude, longitude) and the pages URLs. It will then build a JSON file to be send to the client side. The JSON structure illustrated in the JSON as shown in Figure (3.8). The client side will receive the JSON file and display the results in the map as a marker with

the location title as shown in Figure (3.2). The user able to click the location point to view the location abstract or go to the location DBpedia web page.

For informing the user about the ranking to fetched location according to the user interest expressed by his entered search keywords snippets and distance between the user and the location, a group of one to five starts is placed next to each displayed location on the application map. Those starts represent the result of the ranking equation illustrated in the previous section 3.6.5.

{"Locations":[

{"No":1, "lat":31.504203, "long" : 34.464467, "URL": " http://dbpedia.org/resource/Great_Mosque_of_Gaza","Title":" المسجد العمري (الكبير)","Abstract":".…….الجامع العمري الكبير هو المسجد الأكبر والأقدم في قطاع غزة، ويقع في مدينة غزة القديمة،"}

{"No":2, "lat":31.508056, "long" : 34.463347, "URL": " http://dbpedia.org/resource/Sayed_al-Hashim_Mosque","Title":" مسجد السيد","Abstract":" هاتم، مسجد السيد هاشم يعتبر مسجد السيد هاشم من أقدم مساجد غزة ، وأتقنها بناء ، ويقع في حي الدرج في المنطقة الشمالية لمدينة غزة القديمة ، ويبعد عن المسجد العمري مسافة……."}

{"No":3, "lat":31.504100, "long" : 34.468900, "URL": " http://dbpedia.org/resource/Ibn_Uthman_Mosque","Title":" جامع ابن عثمان "," Abstract":" الجامع هذا ويعتبر فلسطين، جنوب غزة مدينة في الشجاعية حي في يقع عثمان ابن جامع أحد أكبر المساجد الأثرية ونموذج رائع للعمارة المملوكية بعناصرها المعمارية والزخرفية . يقع في حي الشجاعية —شارع السوق، ويعتبر هذا الجامع أحد أكبر المساجد الأثرية…."}   ]}

Figure (3.8): illustrate the JSON structure of the results sends from the server side to the client side.

3.7 Summary

In this chapter we explained the system structure, how it will exploit the LOD and how it will present the results of the LOD DBpedia on top of google maps. Furthermore, we explained how the system is built have two sides, client side and server side. The client side is responsible for gathering client search keywords and GPS location then send the request to the server side using JSON file, finally the client receive the fetched results from the server and display for the user on a google map. The server side resides on an internet connected machine and have reachability to the DBpedia and google services. This server side have six modules. The JSON Processing Module is responsible

for reading and processing the JSON file sent from the client. The DBpedia Based Locator Module is responsible for receives the GPS location from the JSON Processing Module and query the DBpedia for all places around the user location. The Query Preprocessing Module performs basic NLP techniques on the search query entered by the user. The Query Expansion Module fetches the snippets words from google services based on the search keywords generated by Query Preprocessing Module. The Ranking Algorithm Module balances the results importance to the user based on the distance of the place and the user and the user search keywords. The Results Builder module receives the ranked results from the Ranking Algorithm Module, generates a JSON file and sends it to the client.

Chapter 4
Evaluation

Chapter 4   Evaluation

4.1 Introduction

Internet recommendation services means that the user is looking for a service or a piece of information about a topic or a place he needs. The user does that by inputting a search query which consists of words into a recommendation system. The system then fetches the results and presents them to the user.

Evaluating mobile-based recommendation system should consider not only the accuracy of results, but also other factors related to the mobile characteristics. These factors may include the context information, user profile, device type and weather conditions. For example, the position of the user should be considered when searching for nearby restaurants.

Our mobile based recommendation system focuses of places' information which are obtained from the open Source Database DBpedia. In the system the user makes a query on the system by inserting a search query, while the system automatically gets the user location coordinates. Then, the system enquire DBpedia for locations that matches the user query and are close as possible to the detected user location. . Afterwards, results are ranked and displayed to the user according to the relevance and importance. In this Chapter we present and discuss the evaluation process we carried out to evaluate the search results obtained from our system. We focus on evaluating the following factors:

First, the accuracy of the obtained results: The recommendation system aims at retrieving locations that best matches with the user query. Accuracy of retrieved locations are determined based on the relatedness to the input query and the proximity of locations to the user's position. For example, when a user searches for the phrase: "sea food", relevance of retrieved locations is determined based on whether related services are available in the retrieved location, e.g. providers of sea food, as well as the extent to which the location is close to the user's position.

Second, the efficiency of the search service is assessed by measuring the time consumed to perform the search process.

4.2 Experimental Setting

In this section, the experimental settings are explained in detail: the used dataset in the system evaluation is first presented. Afterwards, the used evaluation metrics are explained.

4.2.1 Evaluation Dataset

In general, location-based search services can be best evaluated by being used in practice: Users in different geographical locations should use the system to search for locations of different types. Then, results can be assessed by their relatedness to input queries and the proximity to users' positions.. However, we could not apply this approach to our case due to the following reasons:

First, few geographical locations are currently defined in DBpedia. Based on our assessment, we could find only forty-one locations in Gaza that have reference pages in DBpedia. With such limited number of DBpedia information, it was difficult to assess our system in Gaza.

Instead, we handcrafted a test set. consisting of 41 instances that aim at searching for geographical locations in three Arab regions: Gaza, Cairo and Jerusalem. Each instance consists of a search query and an arbitrary latitude longitude point (see Table 4.1 for sample instances of our dataset). The test set can be downloaded from the URL https://goo.gl/Xw1nFX Instead of having a real user physically located in a specific area, the given latitude longitude point resembles the position of the user when searching for the corresponding query instances. Search queries and position information are selected based on the following criteria:

First, we decided to limit our queries instances to cover places in three geographical areas that are Gaza, Jerusalem and Cairo. These three locations were particularly selected because we have human subjects who have experience in the details of these areas and who could help us in assessing retrieved results with respect to both relevance and proximity.

Second, all selected queries target locations that have corresponding URIs in DBpedia. This is necessary because our system relies primarily on DBpedia to retrieve locations. Of the 41 selected queries, 15 queries refer to locations in Gaza, 13 queries refer to locations in Cairo, while 13 queries refer to locations in Jerusalem. Table 4.1 shows sample instances of our dataset, while the query set is shown in Appendix A.

Third, location coordinates associated with queries were carefully chosen to be within reasonable proximities to the selected locations.

As these coordinates resemble the user's location when using the mobile-based system, it is assumed that the user can be located in major streets and squares in the selected areas. These locations were determined with the help of experts who know the areas and could identify main streets and their distance to target locations. The Google maps are also used to discover details of areas and identify coordinates accurately.

Fourth, length of each query varies between 1 to 4 words. This variety is essential to assess how the performance changes as the number of input keywords vary.

Fifth, the dataset we created for the evaluation is different from the test set we used while developing and tuning our system.

Table (4.1): illustrate the sample instances of our dataset

Serial	Latitude	Longitude	Search Query	City/Country
1	31.5131	34.4405	ملعب فلسطين	Gaza, Palestine
2	31.5131	34.4405	مستشفى الشفاء	Gaza, Palestine
3	31.504203	34.464467	سوق الذهب	Gaza, Palestine
4	31.504203	34.464467	قصر الباشا	Gaza, Palestine
5	31.504203	34.464467	المسجد العمري الكبير	Gaza, Palestine
6	31.751167	35.190617	ستاد	Jerusalem, Palestine
7	31.776667	35.234165	باب العامود	Jerusalem, Palestine
8	31.7722	35.228901	كنيسة رقاد السيدة العذراء	Jerusalem, Palestine
9	31.7722	35.228901	بيت الشرق	Jerusalem, Palestine
10	31.7722	35.228901	المصلى المرواني	Jerusalem, Palestine
11	30.029444	31.261389	قلعة صلاح الدين	Cairo, Egypt
12	30.051167	31.297001	ستاد	Cairo, Egypt
13	30.02	31.299999	جبل المقطم	Cairo, Egypt
14	30.066389	31.227501	دار الكتب والوثائق	Cairo, Egypt
15	30.015479	31.239252	المتحف القبطي	Cairo, Egypt

4.2.2 Evaluation Process

For the assessment of reliability, we ran our recommendation system over the dataset of the user search query with his\here GPS location and recorded the results which are obtained from the DBpedia places webpages. Our system ranked the results based on the relatedness to the user search query and the distance to the user's GPS location. Results were then given to human experts to assess them. Each expert had a good experience in one geographical so that he could assess the relevance of locations retrieved from DBpedia. Experts were asked to rank results using a scale from 1 to 5, where 5 means most relevant while 1 means least relevant.

To illustrate how the system's output looks like, and how it was rated by the expert, Tables 4.2 and 4.3 show results of two search queries: "برج القلعة" (latitude = 31.776112 , longitude = 35.227779) in Jerusalem, and "المسجد العمري الكبير" (latitude = 31.504203, longitude = 34.464467) in Gaza . Results in these tables are ordered according to

ranking given by the system so that top ranked locations are shown first. The first column presents DBpedia URIs of retrieved locations. The second column presents the DBpedia label. The third column presents the Snippet frequent in the abstract of the DBpedia location page. The fourth column shows the distance between the retrieved location and the target location in the query. For example, the first retrieved location in Table 4.2 is about 1316 meters far from the target location, i.e. "برج القلعة". The fifth column shows the rate estimated by the system, the last column contains the expert rates. Note that the rate estimated by the system is a normalized value that ranges from 0 to 1. To make the system's rate comparable with the expert's rate. We mapped the system's rates to values from 1 to 5. This mapping is done by splitting the scale from 0 to 1 into 5 intervals, each of which represents a step of 0.2. Boundaries of these intervals are then mapped to values from 1 to 5, and results within intervals are approximated accordingly.

Table 4.3 describe a results returned from the query with a parameters user search query is "المسجد العمري الكبير" and user GPS location, Palestine, the first column contains the DBpedia webpage URL, the second column contains the page label, the third column contains the snippet word frequent in the page abstract (I did not understand the third column) the last column contains the bag rank which is calculated from the system, The last column contains the expert rates.

Table (4.2): illustrate the results returned from the search service for the user query "برج القلعة" and user GPS location (latitude = 31.776112 , longitude = 35.227779) in Jerusalem

Subject	Label	Count	Distance	System rate	Expert rate
http://DBpedia.org/resource/Jerusalem	القدس@ar	0.5428008	1316.39566	1	4
http://DBpedia.org/resource/Muristan	مورستان@ar	0.445137398	238.1344864	0.8	3
http://DBpedia.org/resource/Jaffa_Gate	باب الخليل@ar	0.398278468	43.75211738	0.8	3
http://DBpedia.org/resource/Tower_of_David	برج القلعة@ar	0.376783068	2.388533715	0.8	5
http://DBpedia.org/resource/Orient_House	بيت الشرق@ar	0.2588406	1437.699954	0.6	3
http://DBpedia.org/resource/Dung_Gate	باب المغاربة@ar	0.199697368	599.0772743	0.6	3
http://DBpedia.org/resou	جبل	0.173348787	496.3182824	0.4	2

45

Subject	Label	Count	Distance	System rate	Expert rate
rce/Mount_Zion	صهيون@ar				
http://DBpedia.org/resource/Solomon's_Stables	المصلى المرواني@ar	0.182895527	862.1251339	0.4	3

Table (4.3): illustrate the results returned from the search service for the user query "المسجد العمري الكبير" and user GPS location (latitude = 31.504203, longitude = 34.464467) in Gaza

Subject	Label	Count	Distance	Rank	Expert rate
http://DBpedia.org/resource/Great_Mosque_of_Gaza	المسجد العمري الكبير (غزة)@ar	0.403071	3.141883	.8	5
http://DBpedia.org/resource/Gaza_Governorate	محافظة غزة@ar	0.378626	2226.824	.6	4
http://DBpedia.org/resource/Sayed_al-Hashim_Mosque	مسجد السيد هاشم@ar	0.186285	446.8897	.6	4
http://DBpedia.org/resource/Ibn_Uthman_Mosque	جامع ابن عثمان@ar	0.157065	419.9495	.6	4

4.3 Evaluation Metrics

To evaluation our results with respect to the rankings given by experts, two metrics were used. These metrics are: MAP (Mean Average Precision) and Normalized discount cumulative gain (nDCM). These two metrics are commonly used to assess recommendation systems. While MAP is mainly used to assess the precision of retrieved results, nDCM is used to assess the ranking of results, and its correlation to the ranking given by the experts. In the following, each metric is explained in detail with an example.

To evaluate the results we will use tow evaluation metrics, the normalized discount cumulative gain and Mean Average Precision (MAP).

4.4 Results and Discussion

Since we were interested in assessing our search service system, the results of the two evaluation metrics: NDCG and MAP for the 41 user search query are summarized in the Table 4.4, while the full results is shown in Appendix B.

Table (4.4): summarized the results evaluation for returned from the proposed solution

Geographical Area	No of queries	nDCG accuracy	MAP Accuracy
Locations in Gaza	15	91.1262429%	76.0585223%
Locations in Jerusalem	13	93.1827625%	83.2797389%
Locations in Cairo	13	89.7164429%	74.2487935%
All locations	41	91.377%	77.7777%

Results evaluation are presented in Table 4.4 for 41 user search query in three cities namely Jerusalem - Palestine, Gaza - Palestine and Cairo - Egypt It achieved (91.377%) accuracy with SD 0.337 using Normalized Discount Cumulative Gain and it achieved 77.77% with SD 0.029 using MAP metric. As shown in table 4.7.

The above results show that MAP value was lower than the nDCG value. This result indicates that the system achieved better in ranking results than in generating accurate results.

Source of Errors:

We inspected the results thoroughly to identify the main sources of errors. Errors can be classified into the following categories based on the source of errors:

Errors due to Google search API: As mentioned in Section 3.5.4, Google search API was used to expand the search query by using words from Google snippets. Google search API was used to retrieve top search snippets. However, many of the returned snippets contained words that are not related to the input keywords. This has cause the system to retrieve wrong or unrelated results from DBpedia. For example, for the user search keyword: "القدس", the Google search API returned many results related to news articles. These news articles contained words that are not of interest to the user such as "موقع, اخباري   يتناول , منوع , الشؤون , العربية , عامة , والشؤون , الخليجية , خاصة , نتبنى , ولكن , نطرحها".. In another example, the user search keywords "كلية العلوم التطبيقية", the Google search API returned in the first rank the snippet words: "تأكد , ادخال , رقمك , الجامعي   , صحيح , كلمة   , المرور , وحالة , الاحرف , انتبه , لزر , الحروف , الكبيرة , لتبقى , تواصل , تابعنا , صفحة , الفيسبوك ,".. In another example, the user search keyword "مسجد ", the Google search

API returned in the first rank the snippet words "الخلافة , الإسلامية , الخلافة , الراشدة , الخلافة , الأموية , الخلافة , العباسية , الخلافة , القرطبيَّة , المسجد , أول , مبنى , تشهده , المدينة , المنورة , للدولة , الأولى , العاصمة , الإسلامية , مباشرة , وصول , النبي , محمد , الصلاة , والسلام , مهاجرا , مكة , شكل , المسجد , إحدى" in this case the user wants the building but the snippet words refer to the Islamic empires along the history and the messenger Mohamed. While the user was interested in the locations related to search queries, unrelated words from search snippets results in many invalid results. Note that our search approach relies on the assumption that the DBpedia page is relevant only if its abstract contains frequent words from search snippets.

Error dues to the DBpedia website : many geographical location DBpedia spaces, have a DBpedia website abstract which contains other query unrelated information, such as historical information about the place name, this information contains a lot of snippet words, which gives the DBpedia website hi frequency words. These websites with high frequency snippets will have a high rank in the results but it is not related for the user search keyword. For example the DBpedia website "مورستان" it's URL is "http://DBpedia.org/resource/Muristan" a part of its abstract is " اسم المستشفى " "كانت المستشفيات في ذلك الوقت على قدر عال بيمارستان"، وهى كلمة أصلها فارسي تعنى " محل المريض للغاية من التنظيم والترتيب والنظافة حتى أن الحديث عنها يكاد يكون أشبه بأفلام الخيال العلمي، خصوصا عندما تعلم أن الحضارة الإسلامية سبقت أوروبا في تأسيس وإنشاء المستشفيات بتسعة قرون. فقد أُسِّس أول مستشفى إسلامي في عهد الخليفة الأموي الوليد بن عبد الملك في دمشق ، وكان هذا المستشفى متخصصًا في الجذام، وأنشئت بعد ذلك المستشفيات العديدة في العالم الإسلامي، وبلغ بعضها شأوًا عظيمًا؛ حتى كانت هذه المستشفيات تُعدّ قلاعًا للعلم والطب، وتُعتبر من أوائل الكليات والجامعات في العالم " , this abstract contains a lot information about the Islamic Khelafeet and the eras since first hospitals built in that period and comparison between them and the European civilization, although this palace is The Muristan (from Persian Bimārestān meaning "hospital") is a complex of streets and shops in the Christian Quarter of the Old City of Jerusalem. In other example, the abstract of a DBPeia website contained little information so it well not get good result in word frequency calculation proses like the page "دار الأوبرا الخديوية" , "http://DBpedia.org/resource/Cairo_Opera_House" it have just one paragraph in the abstract, this made it hard to find in the search.

Errors due to the Arabic DBpedia website weakness: there is a lot of places in the Arab word has no DBpedia resources. for example, Jordanian capital Amman has less than ten DBpedia website for palaces and the Saudi Arabia's capital Riyadh has less than five DBpedia website for palaces and the cites Medina monawara Makkah mokarama in Saudi Arabia's each city has less than twenty DBpedia website for palaces. but all queries in our dataset should have DBpedia pages.

4.5 Time Efficiency

To assess the efficiency of the system, we measured the execution time of the 41 user search queries. The specifications of the machine used in the evaluation process is shown in Table 4.5.

Table 4.5: illustrate the execution times for the 41 user queries

Processor Type	Intel® core(TM)2 Duo CPU
Processor Clock Speed	2.20 Giga Hertz
Installed Memory	3 Giga Byte
Operating System	Windows 7 Ultimate
System Type	32 bit operating system

In order to validate the results, the time efficiency test has been run two times, and gave the results as summarized in table 4.6.

The average execution time for the 41 user query was 8.24 and 6.27 seconds in run 1 and 2 respectively. Standard deviation was 3.24 and 2.5 in run 1 and run2 respectively. The minimum execution time for any text was 3 and 2 seconds and the maximum execution time was 19 and 11 seconds.
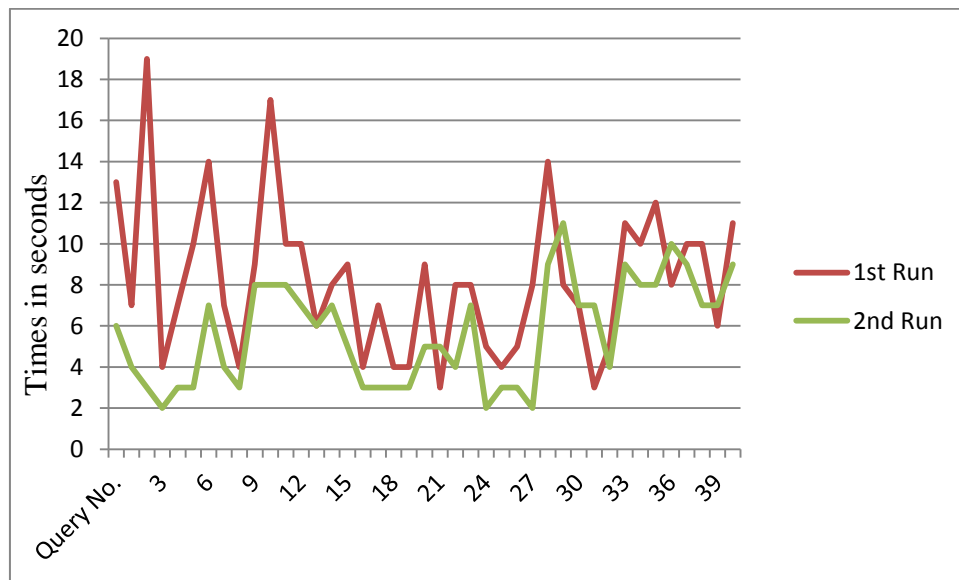

Figure (4.1): Execution Time for the 41 user query over two tests run

Table (4.6): Execution Time

	1st Run	2nd Run
Average Execution Time by seconds	8.24	6.27
Minimum Execution Time by seconds	3	2
Maximum Execution Time by seconds	19	11
Standard Deviation	3.24	2.5

From the results of the time efficiency test, it is clear that time varies in every single query in the two test runs. After investigating this issue and ruling out the fixed factors which are, the machines and program and query. It was clear the internet connection reliability is of the main reasons for this variations. Specially the use of temporary server and temporary infrastructure. In addition, the query expansion process, which aims to extract keywords from Google's search snippets, also consumes significant time.

In general, the prototype implementation of our approach gave priority to the performance of the recommendation algorithm in terms of precision and ranking. Time efficiency was not gave great attention, and will be further investigated in our future work. We think that time efficiency can be improved by exploiting parallel processing as in the case of traditional search engines.

4.6 Configuring the Ranking Algorithm

As explain in Section 3.6.5, the ranking of retrieved locations is based on two factors: the relevance to the user query and the proximity to the user's location. In Equation 3.2, alpha($\propto$) and beta (β) are used to control the weights of the relevance and the proximity factors. We aimed to explore how the performance can be affected by changing alpha and beta. Since alpha and beta both sum to one, we assessed only the relation between alpha and the performance, in terms of MAP and nDCG. We changed alpha from 0 to 1 with a step of 0.2. Table 4.7 and Figure 4.2 show the average MAP and nDCG values calculated for each alpha. Results show that the best performance is achieved when alpha is equal to .7, and hence beta is equal to .3. After this values, performance started degrading. Therefore, we chose alpha=.7 And beta=.3 For our experiment.

Table (4.7): illustrate the results evaluation for returned from the proposed solution

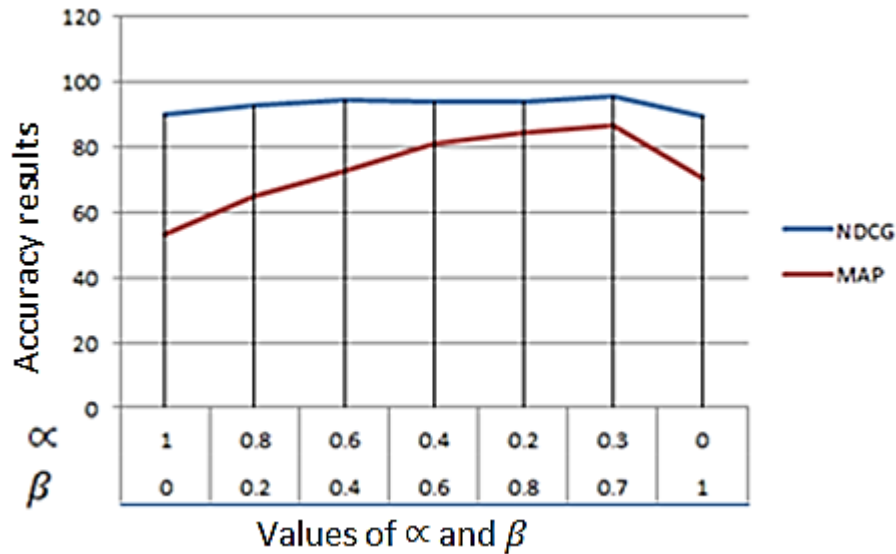| #	∝	β	NDCG	MAP
1	0	1	90.21478	53.06786
2	0.2	0.8	92.7805	65.15234
3	0.4	0.6	94.53898	72.96786
4	0.6	0.4	93.79102	81.02046
5	0.8	0.2	94.19766	84.54216
6	1	0	89.37487	70.59751
7	0.7	0.3	95.38463	86.56566

Figure (4.2): Illustrate the average MAP and nDCG values calculated

4.7 Summary

This chapter presented the evaluation of the approach. It also discussed the results and the sources of resulting errors (the system assessed using the Normalized Discount Cumulative Gain).

We formulated a dataset of 41 user search query with GPS location to evaluate the system results compared to a human evaluation rank. we use Normalized Discount Cumulative Gain to evaluate the system the results indicated that our system achieves results with (91.33%) accuracy and 0.337 standard deviation. The time efficiency test is carried out twice the results are execution time average was 8.24 and 6.27 seconds and the standard deviation was 3.24 and 2.5 respectively in the runs.

51

Chapter 5
Conclusions and Future Work

Chapter 5   Conclusions and Future Work

In the work we designed a recommendation system that exploits the LOD to present recommendations relevant to the user's needs. The use of LOD aims to overcome the shortage of information covering some geographical areas such as the Gaza strip. Unlike other common mapping services, which relies on categorized places, this system supports the use of search keywords and analyses the content of DBpedia resources in order to find most related locations. The recommendation system tries to present the results in an easy to understand way on the map. A ranking algorithm is proposed to rank location according to the user interest.

The system consists of two sides, the client side and the server side. Bothe sides communicate over internet web service using JSON techniques. The client side installed in the mobile phone use the GPS module in the mobile to get the user GPS location and create A JSON file, which contains the user GPS location (latitude, longitude) with the user search keyword and send the JSON file over to the server side. The server side receives the JSON file. The server side six modules process the received JSON file as follow. 1) JSON processing: This module proses the JSON file to split the GSP (latitude, longitude) and send them to the DBpedia based locator finder module, and sends the user search keyword Query preprocessing module. 2) DBpedia based locator finder: This module query the DBPedia to get the places DBPedia web pages around the user location. 3) Query preprocessing: This module performs basic NLP techniques on the user search keyword, the NLP techniques is performed on three steps first is tokenization the second is light stemming and the third is stop word removal. 4) Related Keyword: In this module, we use the Google search service API to get the snippet word for the important user search key word, the snippet word send to the Query preprocessing module to be processed. 5) Ranking algorithm: This module receives data from the DBpedia based locator finder, and receives the processed user search keyword snippet words from the Query preprocessing module; in the Ranking algorithm, we apply the semantic search to arrange the results from the most important to the least. 6) The Results builder: this module receives the results from the Ranking algorithm to build the JSON and send it to the client. At the end, the client side present the locations in a Google map as a mark points in the map.

The work in this thesis was evaluate by two kind of testing. First is the reliability, second is Time Efficiency. The reliability testing consists of two metrics the first is Mean Average Precision (MAP) and the second is Normalized Discount Cumulative Gain (nDCM). The system is assessed over a data set of 41 user query form different places and towns, the system achieved on MAP metric 77.77% with SD 0.029 and in

Normalized Discount Cumulative Gain 91.33% accuracy with SD 0.337 using, the Time Efficiency achieved 8.24 Average Execution Time with 3.24 Standard Deviation.

Several tests have been performed on the system for evaluation. The performance test used a sample dataset consisting of locations defined in DBpedia. Two metrics are used to assess precision and ranking of results: the first is Mean Average Precision (MAP) and the second is Normalized Discount Cumulative Gain (nDCM). The system achieved on MAP metric 77.77% with SD 0.029 and in Normalized Discount Cumulative Gain 91.33% accuracy with SD 0.337 using. The time efficiency test is carried out twice the results are execution time average was 8.24 and 6.27 seconds and the standard deviation was 3.24 and 2.5 respectively in the runs.

The main contributions in this work is the use of DBpedia as source of the data to provide the user with information about proximate locations relevant to his/her needs. The user's location is automatically detected from the GPS module in the mobile phone. A ranking algorithm was also explained to make balance between proximity and relevance of locations.

5.1 Future work
Since this work is trying to cover uncovered areas by common map services, it is clear that a LOD sources of information must be enriched, therefore upgrading the system and enabling it to update and add information to the DBpedia our open source if information will be viable.

The system can be upgraded to add the weather conditions to the user context, and recommend weather suitable places.

It is viable to build a cumulative user profile and store user data and his interests, further his choosing while using the recommendation system. Such upgrade will make the system self-learning and provide more a curate results in the future.

The security is beyond the scope of this work, it highly recommended to evaluate the security of the system and the privacy of the user.

The time efficiency posed an issue in this work, a future work on improving the time efficiency is highly recommended specially by exploiting parallel processing.

A new approach for recommendation system is suggested, the results of the new approach to be compared with this approach.

References

References

Adomavicius, Gediminas, Sankaranarayanan, Ramesh, Sen, Shahana, & Tuzhilin, Alexander. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS), 23*(1), 103-145.

Adomavicius, Gediminas, & Tuzhilin, Alexander. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering, 17*(6), 734-749.

Adomavicius, Gediminas, & Tuzhilin, Alexander. (2011). Context-aware recommender systems *Recommender systems handbook* (pp. 217-253): Springer.

Agichtein, Eugene, Brill, Eric, & Dumais, Susan. (2006). *Improving web search ranking by incorporating user behavior information.* Paper presented at the Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval.

Ahlqvist, Emil. (2015). Implementing a Resum eDatabase with Online Learning to Rank.

Auer, Sören, Bizer, Christian, Kobilarov, Georgi, Lehmann, Jens, Cyganiak, Richard, & Ives, Zachary. (2007). *Dbpedia: A nucleus for a web of open data*: Springer.

Bäck, Asta, Vainikainen, Sari, Södergård, Caj, & Juhola, Helene. (2003). *Semantic Web Technologies in Knowledge Management.* Paper presented at the ELPUB.

Becker, Christian, & Bizer, Christian. (2008). DBpedia Mobile: A Location-Enabled Linked Data Browser. *LDOW, 369*.

Bellotti, Victoria, Begole, Bo, Chi, Ed H, Ducheneaut, Nicolas, Fang, Ji, Isaacs, Ellen, . . . Price, Bob. (2008). *Activity-based serendipitous recommendations with the Magitti mobile leisure guide.* Paper presented at the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.

Bizer, Christian, Lehmann, Jens, Kobilarov, Georgi, Auer, Sören, Becker, Christian, Cyganiak, Richard, & Hellmann, Sebastian. (2009). DBpedia-A crystallization point for the Web of Data. *Web Semantics: science, services and agents on the world wide web, 7*(3), 154-165.

Bouidghaghen, Ourdia, Tamine-Lechani, Lynda, & Boughanem, Mohand. (2009). Dynamically personalizing search results for mobile users *Flexible Query Answering Systems* (pp. 99-110): Springer.

Bouneffouf, Djallel. (2013). Optimizing an Utility Function for Exploration/Exploitation Trade-off in Context-Aware Recommender System. *arXiv preprint arXiv:1303.0485*.

Bouneffouf, Djallel, Bouzeghoub, Amel, & Gançarski, Alda Lopes. (2012). *Following the User's Interests in Mobile Context-Aware Recommender Systems: The Hybrid-e-greedy Algorithm.* Paper presented at the Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on.

BRAMSTÅNG, ALBIN, & JIN, YANLING. (2015). Constructing a Context-aware Recommender System with Web Sessions.

Chen, Hung-Hsuan, Gou, Liang, Zhang, Xiaolong, & Giles, Clyde Lee. (2011). *Collabseer: a search engine for collaboration discovery.* Paper presented at the Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries.

Chen, Hung-Hsuan, Ororbia, II, Alexander, G, & Giles, C Lee. (2015). ExpertSeer: a Keyphrase Based Expert Recommender for Digital Libraries. *arXiv preprint arXiv:1511.02058*.

Clarke, Charles LA, Kolla, Maheedhar, Cormack, Gordon V, Vechtomova, Olga, Ashkan, Azin, Büttcher, Stefan, & MacKinnon, Ian. (2008). *Novelty and diversity in information retrieval evaluation.* Paper presented at the Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval.

Corlosquet, Stéphane, & Clark, Lin. (2011). The Semantic Web, Linked Data and Drupal, Part 2: Combine linked datasets with Drupal 7 and SPARQL Views. *developerWorks, IBM (May 2011). url: http://www. ibm. com/developerworks/web/library/wa-datasets*.

Danylenko, Antonina, & Löwe, Welf. (2012). *Context-aware recommender systems for non-functional requirements.* Paper presented at the Proceedings of the Third International Workshop on Recommendation Systems for Software Engineering.

DBpedia. (2017, 21/03/2017). 2017, from http://wiki.dbpedia.org/about

De Pessemier, Toon, Dooms, Simon, & Martens, Luc. (2014). Context-aware recommendations through context and activity recognition in a mobile environment. *Multimedia Tools and Applications, 72*(3), 2925-2948.

de Spindler, Alexandre, Norrie, Moira C, Grossniklaus, Michael, & Signer, Beat. (2006). Spatio-temporal proximity as a basis for collaborative filtering in mobile environments.

El-Radie, Omar Salah, & Alagha, Iyad M. (2015). SPARQL2AL: Translating SPARQL Queries to Arabic Language. *Islamic University-Gaza*.

Felfernig, Alexander, Isak, Klaus, Szabo, Kalman, & Zachar, Peter. (2007). *The VITA financial services sales support environment.* Paper presented at the PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE.

Gavalas, Damianos, Konstantopoulos, Charalampos, Mastakas, Konstantinos, & Pantziou, Grammati. (2014). Mobile recommender systems in tourism. *Journal of Network and Computer Applications, 39*, 319-333.

Ge, Yong, Xiong, Hui, Tuzhilin, Alexander, Xiao, Keli, Gruteser, Marco, & Pazzani, Michael. (2010). *An energy-efficient mobile recommender system.* Paper presented at the Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining.

Ghazanfar, Mustansar Ali, Prügel-Bennett, Adam, & Szedmak, Sandor. (2012). Kernel-mapping recommender system algorithms. *Information Sciences, 208*, 81-104.

Hjortdal, Vibeke E, Redington, AN, De Leval, MR, & Tsang, VT. (2002). Hybrid approaches to complex congenital cardiac surgery. *European journal of cardio-thoracic surgery, 22*(6), 885-890.

Howland, B. McCulloch WS. Pitts W. and Wall P. 0.(1955) ReRex inhibition by dorsal root interaction. *J. Neuroph_rsioL, 18*.

ibm. (2016). Retrieved 09/02/2016, 2016, from http://www.ibm.com/developerworks/library/wa-datasets

Jafarkarimi, Hosein, Sim, Alex Tze Hiang, & Saadatdoost, Robab. (2012). A naive recommendation model for large databases. *International Journal of Information and Education Technology, 2*(3), 216.

Jain, Vishal, & Singh, Mayank. (2013). Ontology Development and Query Retrieval using Protutf-8. *International Journal of Intelligent Systems and Applications, 5*(9), 67.

Järvelin, Kalervo, & Kekäläinen, Jaana. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS), 20*(4), 422-446.

Jung, Hoill, & Chung, Kyungyong. (2016). Knowledge-based dietary nutrition recommendation for obese management. *Information Technology and Management, 17*(1), 29-42.

Liu, Duen-Ren, & Shih, Ya-Yueh. (2005). Hybrid approaches to product recommendation based on customer lifetime value and purchase preferences. *Journal of Systems and Software, 77*(2), 181-191.

Masthoff, Judith, Mobasher, Bamshad, Desmarais, Michel, & Nkambou, Roger. (2012). *User Modeling, Adaptation, and Personalization: 20th International Conference, UMAP 2012, Montreal, Canada, July 16-20, 2012 Proceedings* (Vol. 7379): Springer.

McCarthy, Luke, Vandervalk, Ben, & Wilkinson, Mark. (2012). SPARQL Assist language-neutral query composer. *BMC bioinformatics, 13*(1), S2.

Melville, Prem, & Sindhwani, Vikas. (2010). Encyclopedia of machine learning: Springer-Verlag, chapter Recommender systems.

Mendes, Pablo N, Jakob, Max, & Bizer, Christian. (2012). *DBpedia: A Multilingual Cross-domain Knowledge Base.* Paper presented at the LREC.

Mitchell, Erik T. (2013). Building Blocks of Linked Open Data in Libraries. *Library Technology Reports, 49*(5), 11-25.

Montaner, Miquel, López, Beatriz, & De La Rosa, Josep Lluís. (2003). A taxonomy of recommender agents on the internet. *Artificial intelligence review, 19*(4), 285-330.

Mooney, Raymond J, & Roy, Loriene. (2000). *Content-based book recommending using learning for text categorization.* Paper presented at the Proceedings of the fifth ACM conference on Digital libraries.

Muntean, Cristina Ioana, Nardini, Franco Maria, Silvestri, Fabrizio, & Baraglia, Ranieri. (2015). On learning prediction models for tourists paths. *ACM Transactions on Intelligent Systems and Technology (TIST), 7*(1), 8.

Musto, Cataldo, Basile, Pierpaolo, de Gemmis, Marco, Lops, Pasquale, Semeraro, Giovanni, & Rutigliano, Simone. (2015). *Automatic Selection of Linked Open Data Features in Graph-based Recommender Systems.* Paper presented at the CBRecSys@ RecSys.

Ostuni, Vito Claudio, Di Noia, Tommaso, Mirizzi, Roberto, Romito, Davide, & Di Sciascio, Eugenio. (2012). Cinemappy: a Context-aware Mobile App for Movie Recommendations boosted by DBpedia. *SeRSy, 919*, 37-48.

Ou, Shiyan, Orasan, Constantin, Mekhaldi, Dalila, & Hasler, Laura. (2008). *Automatic Question Pattern Generation for Ontology-based Question Answering.* Paper presented at the FLAIRS Conference.

Paireekreng, Worapat. (2013). *Mobile content recommendation system for re-visiting user using content-based filtering and client-side user profile.* Paper presented at the Machine Learning and Cybernetics (ICMLC), 2013 International Conference on.

Pan, Jeff Z, & Horrocks, Ian. (2007). Rdfs (fa): connecting rdf (s) and owl dl. *IEEE Transactions on Knowledge and Data Engineering, 19*(2).

Panayiotou, Christoforos, & Samaras, George. (2006). *Mobile user personalization with dynamic profiles: Time and activity.* Paper presented at the On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops.

Panniello, Umberto, Tuzhilin, Alexander, Gorgoglione, Michele, Palmisano, Cosimo, & Pedone, Anto. (2009). *Experimental comparison of pre-vs. post-filtering approaches in context-*

aware recommender systems. Paper presented at the Proceedings of the third ACM conference on Recommender systems.

Park, Moon-Hee, Hong, Jin-Hyuk, & Cho, Sung-Bae. (2007). Location-based recommendation system using bayesian user's preference model in mobile devices *Ubiquitous Intelligence and Computing* (pp. 1130-1139): Springer.

Passant, Alexandre. (2010). *dbrec—music recommendations using DBpedia.* Paper presented at the International Semantic Web Conference.

Prud'hommeaux, Eric, & Seaborne, Andy. (2006). SPARQL Query Language for RDF ", W3C Working Draft 4 October 2006. *SPARQL Query Language for RDF*.

Ramaswamy, Lakshmish, Deepak, P, Polavarapu, Ramana, Gunasekera, Kutila, Garg, Dinesh, Visweswariah, Karthik, & Kalyanaraman, Shivkumar. (2009). *Caesar: A context-aware, social recommender system for low-end mobile devices.* Paper presented at the Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on.

Rawat, Yogesh Singh, & Kankanhalli, Mohan S. (2017). ClickSmart: A Context-Aware Viewpoint Recommendation System for Mobile Photography. *IEEE Transactions on Circuits and Systems for Video Technology, 27*(1), 149-158.

Ricci, Francesco, & Nguyen, Quang Nhat. (2007). Acquiring and revising preferences in a critique-based mobile recommender system. *IEEE Intelligent systems, 22*(3).

Ricci, Francesco, Rokach, Lior, & Shapira, Bracha. (2011). *Introduction to recommender systems handbook*: Springer.

Rodríguez, Horacio, Farwell, David, Farreres, Javi, Bertran, Manuel, Alkhalifa, Musa, Martí, M Antonia, . . . Pease, Adam. (2008). *Arabic wordnet: Current state and future extensions.* Paper presented at the Proceedings of The Fourth Global WordNet Conference, Szeged, Hungary.

Sridevi, M, Rao, R Rajeshwara, & Rao, M Varaprasad. (2016). A Survey on Recommender System. *International Journal of Computer Science and Information Security, 14*(5), 265.

Vagliano, Iacopo, Figueroa, Cristhian, Rocha, Oscar Rodríguez, Torchiano, Marco, Faron-Zucker, Catherine, & Morisio, Maurizio. (2016). *ReDyAl: A Dynamic Recommendation Algorithm based on Linked Data.* Paper presented at the 3rd Workshop on New Trends in Content-Based Recommender Systems co-located with ACM Conference on Recommender Systems (RecSys 2016).

wikipedia. (2017, 21/03/2017). wikipedia. Retrieved 21/03/2017, from https://en.wikipedia.org/wiki/DBpedia

Zhang, Jia-Dong, & Chow, Chi-Yin. (2016). TICRec: A probabilistic framework to utilize temporal influence correlations for time-aware location recommendations. *IEEE Transactions on Services Computing, 9*(4), 633-646.

Zhang, Wei, & Wang, Jianyong. (2015). *Location and time aware social collaborative retrieval for new successive point-of-interest recommendation.* Paper presented at the Proceedings of the 24th ACM International on Conference on Information and Knowledge Management.

Zheng, Vincent W, Zheng, Yu, Xie, Xing, & Yang, Qiang. (2010). *Collaborative location and activity recommendations with gps history data.* Paper presented at the Proceedings of the 19th international conference on World wide web.

Zheng, Yu, Zheng, Wencheng, & Xie, Xing. (2014). Collaborative location and activity recommendations: Google Patents.

Ziegler, Cai-Nicolas, Lausen, Georg, & Schmidt-Thieme, Lars. (2004). *Taxonomy-driven computation of product recommendations.* Paper presented at the Proceedings of the thirteenth ACM international conference on Information and knowledge management.

Appendices

Appendices

Appendix A: illustrate the instances of our dataset

Serial	Latitude	Longitude	Search Query	City/Country
1	31.4979222	34.4369472	كلية العلوم التطبيقية	Gaza
2	31.52583333	34.43055555	ميناي غزة	Gaza
3	31.474722	34.47361	معبر المنطار	Gaza
4	31.5131	34.4405	مسجد	Gaza
5	31.5131	34.4405	الصبرة	Gaza, Palestine
6	31.5131	34.4405	ملعب فلسطين	Gaza, Palestine
7	31.5131	34.4405	مستشفى الشفاء	Gaza, Palestine
8	31.504203	34.464467	سوق الذهب	Gaza, Palestine
9	31.504203	34.464467	قصر الباشا	Gaza, Palestine
10	31.5131	34.4405	تل الهوى	Gaza, Palestine
11	31.5362972	34.465827	الشيخ رضوان	Gaza, Palestine
12	31.504203	34.464467	المسجد العمري الكبير	Gaza, Palestine
13	31.504203	34.464467	كنيسة القديس برفيريوس	Gaza, Palestine
14	31.504203	34.464467	مسجد السيد هاشم	Gaza, Palestine
15	31.5131	34.4405	الجامعة الإسلامية غزة	Gaza, Palestine
16	30.02861	31.259722	مسجد محمد علي	Cairo, Egipt
17	30.015479	31.239252	المتحف القبطي	Cairo, Egipt
18	30.015804	31.235808	كنيس بن عزرا	Cairo, Egipt
19	30.019762	31.253414	قصر خيري باشا	Cairo, Egipt
20	30.019762	31.253414	المتحف الجيولوجي المصري	Cairo, Egipt
21	30.019762	31.253414	متحف أحمد شوقي	Cairo, Egipt
22	30.050699	31.247999	دار الأوبرا	Cairo, Egipt
23	30.045834	31.224445	برج القاهرة	Cairo, Egipt
24	30.029167	31.213055	حديقة الأورمان	Cairo, Egipt
25	30.029444	31.261389	قلعة صلاح الدين	Cairo, Egipt
26	30.051167	31.297001	ستاد	Cairo, Egipt
27	30.02	31.299999	جبل المقطم	Cairo, Egipt
28	30.066389	31.227501	دار الكتب والوثائق	Cairo, Egipt
29	31.751167	35.190617	ستاد	Jerusalem, Palestine
30	31.776667	35.234165	باب العامود	Jerusalem, Palestine

Serial	Latitude	Longitude	Search Query	City/Country
31	31.776667	35.234165	القدس	Jerusalem, Palestine
32	31.776667	35.234165	البلدة القديمة	Jerusalem, Palestine
33	31.751167	35.190617	رياضة	Jerusalem, Palestine
34	31.776112	35.227779	برج القلعة	Jerusalem, Palestine
35	31.7722	35.228901	كنيسة رقاد السيدة العذراء	Jerusalem, Palestine
36	31.7722	35.228901	بيت الشرق	Jerusalem, Palestine
37	31.7722	35.228901	المصلى المرواني	Jerusalem, Palestine
38	31.776236	35.235577	مذبحة الأقصى الأولى	Jerusalem, Palestine
39	31.783611	35.234165	متحف روكفلر	Jerusalem, Palestine
40	31.755909	35.261379	جامعة القدس	Jerusalem, Palestine
41	31.788218	35.229466	قبور السلاطين	Jerusalem, Palestine

Appendix B: illustrate the results evaluation for returned from the proposed solution

#	Subject	Label	Latitude	Longitude	NDCG Results	MAP Results
1	Jerusalem, Palestine	ستاد	31.75117	35.19062	0.910333	0.827438
2	Jerusalem, Palestine	باب العامود	31.77667	35.23417	0.928355	1
3	Jerusalem, Palestine	القدس	31.77667	35.23417	0.978901	0.946781
4	Jerusalem, Palestine	البلدة القديمة	31.77667	35.23417	0.926558	0.810615
5	Jerusalem, Palestine	رياضة	31.75117	35.19062	0.868074	0.411111
6	Jerusalem, Palestine	برج القلعة	31.77611	35.22778	0.942649	0.970486
7	Jerusalem, Palestine	كنيسة رقاد السيدة العذراء	31.7722	35.2289	0.86604	0.581944
8	Jerusalem, Palestine	بيت الشرق	31.7722	35.2289	0.959609	0.821429
9	Jerusalem, Palestine	المصلى المرواني	31.7722	35.2289	0.938179	0.868707
10	Jerusalem, Palestine	مذبحة الأقصى الأولى	31.77624	35.23558	0.919668	0.931796
11	Jerusalem, Palestine	متحف روكفلر	31.78361	35.23417	0.915821	0.920685
12	Jerusalem, Palestine	جامعة القدس	31.75591	35.26138	0.988484	0.95
13	Jerusalem, Palestine	قبور السلاطين	31.78822	35.22947	0.971088	0.785374
14	Gaza, Palestine	كلية العلوم التطبيقية	31.49792	34.43695	0.82908	0.572222
15	Gaza, Palestine	ميناي غزة	31.52583	34.43056	0.968948	0.583333
16	Gaza, Palestine	معبر المنطار	31.47472	34.47361	0.800418455	0.25
17	Gaza, Palestine	مسجد	31.5131	34.4405	0.960139	0.916667
18	Gaza, Palestine	الصبرة	31.5131	34.4405	0.809844	0.625
19	Gaza, Palestine	ملعب فلسطين	31.5131	34.4405	0.813037	0.634286
20	Gaza,	مستشفى الشفاء	31.5131	34.4405	0.905318	0.743333

#	Subject	Label	Latitude	Longitude	NDCG Results	MAP Results
	Palestine					
21	Gaza, Palestine	سوق الذهب	31.5042	34.46447	0.968429	0.732738
22	Gaza, Palestine	قصر الباشا	31.5042	34.46447	0.982792	0.895685
23	Gaza, Palestine	تل الهوى	31.5131	34.4405	0.935291	0.895685
24	Gaza, Palestine	الشيخ رضوان	31.5363	34.46583	0.951517	0.912925
25	Gaza, Palestine	المسجد العمري الكبير	31.5042	34.46447	0.969492	1
26	Gaza, Palestine	كنيسة القديس برفيريوس	31.5042	34.46447	0.895127	0.755159
27	Gaza, Palestine	مسجد السيد هاشم	31.5042	34.46447	0.978773	0.788889
28	Gaza, Palestine	الجامعة الإسلامية غزة	31.5131	34.4405	0.919588	0.852857
29	Cairo, Egypt	مسجد محمد علي	30.02861	31.25972	0.874515	0.590278
30	Cairo, Egypt	المتحف القبطي	30.01548	31.23925	0.880294	0.65
31	Cairo, Egypt	كنيس بن عزرا	30.0158	31.23581	0.968846	0.877381
32	Cairo, Egypt	قصر خيري باشا	30.01976	31.25341	0.953689	0.9
33	Cairo, Egypt	المتحف الجيولوجي المصري	30.01976	31.25341	0.955895	0.844104
34	Cairo, Egypt	متحف أحمد شوقي	30.01976	31.25341	0.95164	0.810516
35	Cairo, Egypt	دار الأوبرا	30.0507	31.248	0.848936	0.672619
36	Cairo, Egypt	برج القاهرة	30.04583	31.22445	0.906117	0.728704
37	Cairo, Egypt	حديقة الأورمان	30.02917	31.21306	0.863698	0.772619
38	Cairo, Egypt	قلعة صلاح الدين	30.02944	31.26139	0.849747	0.737585
39	Cairo, Egypt	ستاد	30.05117	31.297	0.980875	1
40	Cairo, Egypt	جبل المقطم	30.02	31.3	0.852831	0.416667

#	Subject	Label	Latitude	Longitude	NDCG Results	MAP Results
41	Cairo, Egypt	دار الكتب والوثائق	30.06639	31.2275	31.2275	0.776055
			Summation		37.4646893	31.63749
			Total percent		91.377291%	77.7777%