Dean of Graduate Studies
Al-Quds University

Image Retrieval System Based on Arabic Ontology

Samah S. Kareem

M.Sc. Thesis

Jerusalem-Palestine

1435 / 2014

Image Retrieval System Based on Arabic Ontology

Prepared By: Samah S. Kareem

B.Sc.: Computer Science-Al-Quds Open University-Palestine

Supervisor: Dr. Badai Sartwai

A thesis Submitted in Partial fulfillment of requirements for the degree of Master of Computer Science /Department of Computer Science / Faculty of Graduate Studies – Al-Quds University.

1435 / 2014

Deanship of Graduate Studies

Computer Science Department

Thesis Approval

Image Retrieval System Based on Arabic Ontology

Prepared By: Samah S. Kareem
Registration No: 20913088

Supervisor: Dr. Badai Sartawi

Master thesis submitted and accepted. Date: / /2014
The names and signatures of the examining committee members are as follows:

1- Head of Committee: Dr. Badie Sartawi / Signature: ……..…..
2- Internal Examiner: Dr. Raid Zaghal / Signature: ……..……..
3- External Examiner: Dr. Mohammed Maree / Signature:……..……

Jerusalem – Palestine
1435 / 2014

Dedication

This work is dedicated…

To my mother soul, for her unlimited love, inspiration and encouragement.

To my beloved husband, without his caring support it would not have been possible…

To my sons and daughter …. For my wasting time in studying without taking care for them.

To my brothers, sisters, friends and colleagues…

Samah S. Kareem

27/8/2014

Declaration

I certify that this thesis submitted for the degree of Master, is the result of my own research, except where otherwise acknowledged, and that this study (or any part of the same) has not been submitted for a higher degree to any other university or institution.

Signed……..

Samah S. Kareem

Date: / /2014

Acknowledgments

First and foremost Praise be to the Almighty Allah, our gad, our creator and keeper , for His graces and blessings throughout all my life.
Without Him, I believe that I couldn't do anything.

My sincere thanks for my supervisor Dr. Badie Sartawi, for his sincere efforts, interest and time he have kindly spent to guide my research.

I am extremely grateful and indebted to my colleague Bassam Darass for his help during my thesis process.

I am very grateful to all professors at Al-Quds University-Computer Science department,
for the time they spent to teach me.

Finally, and most importantly, I would like to thank my husband and children. Their support, encouragement,
quiet patience and unwavering love were undeniable.

Image Retrieval System Based on Arabic Ontology

Prepared By: Samah S. Kareem

Supervisor: Dr. Badai Sartwai

Abstract

Since the beginning of the nineties, ontologies have become more and more popular research topic investigated by several researchers, because ontologies are so popular in large part due to what they promise and give; a shared and common understanding of a domain that can be communicated between people and application systems. This concept of using ontology isn't examined in many Arabic computation systems.

Accordingly, in this work, we attempted building an Arabic ontology-based image retrieval system, this system converts natural language queries into machine-understandable formats based on the exploited Arabic ontology.

We chose images as our web content because images are an important source for content on the web. The amount of images information is rapidly rising due to digital cameras and mobile telephones equipped with such devices. This thesis discussed this problem by explaining what happens when the end-user is faced with a repository of images whose content is complicated and partly unknown to the user and how the Semantic Web provides new insights into the image retrieval problem, by developing techniques to retrieve the images based on ontology. We examined and evaluated this approach problem through building a system that can facilitate image retrieval based on Arabic domain ontology, Arabic morphology and syntax rules. Our last step to emphasize the efficiency of the ontology in improving the information retrieval systems, was comparing the retrieval of images which is conducted by matching the semantic and the structural descriptions of the user query, with the retrieval of images

which is conducted based on relational database, which contains descriptive metadata for images.

نظام استرجاع الصور بناءا على الانطولوجيا العربية

إعداد سماح صابر سليمان كريم

إشراف د. بديع السرطاوي.

الملخص :

منذ بداية التسعينات أصبحت الانطولوجيا الاكثر شعبية في مجالات البحث التي عمل عليها العديد من الباحثين ، بسبب ما وعدت وما قامت باعطاءه وانجازه : كالفهم المشترك والمشاركة للمجال الذي يتم التواصل به بين الناس ونظم التطبيقات .هذا المفهوم لاستخدام الأنطولوجيا لا زال لا غير مستخدم في كثير من نظم الكمبيوتر العربية.

حاولنا في هذه الاطروحة طرح واختبار نظام لاسترجاع الصور مبني على الانطولوجيا العربية ،الية عمل هذا النظام تقوم على تحويل اللغة الطبيعية للاستعلام الى اللغة الدلالية التي يمكن مطابقتها بسهولة بجذور البيانات المخزنة ، بناءا على المعيار الدلالي على شبكة الإنترنت لوصف الدلالات الذي يدعم التمثيل الدلالي لقراءة الأجهزة وقد قمنا باختيار الصور كمصدر رئيسي للبيانات التي يقوم عليها النظام لأن الصور هي مصدر رئيسي للمحتوى الاليكتروني على شبكة الإنترنت.ومع التزايد السريع في كمية الصور الاليكترونية نتيجة لانتشار الكاميرات الرقمية والتلفونات الذكية التي تدعم هذه الصور.

قمنا أيضا في هذه الاطروحة بمناقشة المشكلة عندما يواجه المستخدم النهائي مستودع من الصور محتواه معقد وغير معروف جزئيا للمستخدم وطرح الرؤى الجديدة التي يقدمها الويب الدلالي في مشكلة استرجاع الصور والتقنيات المطورة لاسترداد الصور باستخدام ألانطولوجيا.

وبغرض عرض هذه المشكلة والعمل على أيجاد حلول لها قمنا باختبار وفحص نتائج من خلال بناء نهج لنظام يسهل عملية استرجاع الصور بناءا على الانطولوجيا العربية باستخدام اسس وقواعد الصرف والنحو الخاصة باللغة العربية.

قمنا بعد ذلك بعمل مقارنة جذرية بين النظامين المبني على الانطولوجيا والأخر المبني على قاعدة البيانات العلائقية للتأكيد على درجة الدقة والكفاءة التي تقوم بتحقيقها الانطولوجيا للأنظمة التي تبنى بالاعتماد عليها...

Table of Contents

List of figures

Number *Page*

List of Tables

Number *Page*

Chapter One

Introduction

The Semantic Web initiative opens new possibilities which the old World Wide Web could not deliver because of; the non semantic nature of HTML and URLs.

It enables accurate relational information retrieval, by characterizing the difference between two or more descriptions of an object in different representations, by bridging the semantic gap between human concepts, and low-level visual features that are extracted from the images.

The Semantic Web provides additional capabilities that enable information sharing, between different resources, which are semantically represented. It consists of a set of standards and technologies that include a simple data model for representing information (RDF), a query language for RDF (SPARQL), a schema language describing RDF vocabularies (RDFS), a language for describing and sharing ontologies (OWL). These technologies together build up the foundation to formally describe, query and exchange information with explicit semantics [1].

Ontologies have been suggested as a way to solve the problem of information heterogeneity by providing formal, explicit definitions of data and reasoning ability over related concepts, that semantic heterogeneity among information sources needs to be resolved to enable meaningful information exchange or interoperation among them. Ontologies are the structural frameworks for organizing information, to be used in computer science to facilitate knowledge sharing and reuse. Since the beginning of the nineties, ontologies have become more and more popular research topic investigated by several researchers, because ontologies are so popular in large part due to what they promise and give; a shared and common understanding of a domain that can be communicated between people and application systems. All research communities

including natural-language processing, and knowledge representation, recently, artificial intelligence, software engineering, biomedical informatics, library science, enterprise bookmarking, and information architecture as a form of knowledge representation about the world or some part of it, by representing knowledge as a hierarchy of concepts within a domain, using a shared vocabulary to denote the types, properties and interrelationships of those concepts.

According to (MPEG-7) [4], Images are an important source of content on the web. The amount of the informative images is rapidly rising due to digital cameras and mobile telephones equipped with such devices. In this work I have tried to find a resolve to the repository of images whose content is complicated and partly unknown to the user, by adopting Semantic Web new insights into the image retrieval problem, in developing new techniques to retrieve the images based on ontologies. We approached this general problem through building a prototype that can facilitate image retrieval based on Arabic domain ontology by using a language phrase stemmer, this system functionality is done through converting a natural language query into matching stem format, based on a semantic-web standard which supports machine readable semantic representation. The confirmation step of our research was comparing the results which were conducted for the retrieval of images based on matching the semantic and structural descriptions with the user query, and the results which were conducted for the retrieval of images based on relational database. We used as a case study Palestine tourism domain to test our prototype by retrieving the historical religious and natural images using the similarity matching and retrieval algorithms.

This thesis contains several research problems for building web service for image retrieval based on Arabic ontology, starting from querying in Arabic natural language, then

stemming the query and finally matching between the query and the relational database based on the Arabic domain ontology according to similarity algorithms, and the integration aspects within this process in particular. This Chapter presents a brief outline of the scope, contribution, motivation of the research objectives for this thesis. Chapter 2 provides the related work and the main concepts which are discussed during this thesis. In Chapter 3 we present the Arabic challenges in computer systems, the Arabic ontology structure design and implement, the Arabic stemmer structure design and implement., In Chapter 4 we explain the differences between the relational database & ontology, In Chapter 5 we describe the implementation methods, tools, algorithms and equations we used to establish our two prototypes. Finally in Chapter 5 we explain the evaluation results we got during practical experiment, conclusion and future work.

1.1. Thesis Scope

An image retrieval system is a computer system for browsing, searching and retrieving images from a large database of digital images. Regarding the WSRP [2], several criteria can be considered in order to classify image retrieval systems:

- User interaction, browsing, typing text or inserting an image that is visually similar to the target image.
- Search performance, how the search engine actually searches. For instance, whether the search is accomplished through the analysis of visual features or through semantic annotations.
- Domain of the search, standalone search engine, only executes a search in a local computer that versus Internet based search engine.

1.2. Problem Definition

- Impaired ability to find desired images depending on Arabic expression, because all search engines work is based on English that they have built which is based on English language rules because of; lack of Arabic ontologies.
- Mismatch a lot of retrieved pictures with required images, because most of traditional search engines is based on image processing techniques whose main drawbacks are low retrieval precision and difficulty to formulate an exact feature query which pays attention to differences between "human users' high level interpretation of the semantics of visual information and the low-level visual features that can be automatically extracted, creating the so called semantic gap, so there is a need of efficient image retrieval systems.

1.3. Contribution

Our contribution lies in the development of two image search engines, one based on Ontology and the other based on relational database; the examination and evaluation of both of them to see the difference in the performance and accuracy. We built the two search engines as prototypes to serve as a back-end of human language query.

The first one is based on the semantic layer which mainly depends on Ontology to make fast and accurate retrieval according to the user query, the other makes the same but according to the relational database.

We built the two systems for Arabic users and queries in Arabic, so the two prototypes basically have been built based on Arabic linguistics and syntax rules.

1.4. Motivation

The vision of the Semantic Web (Berners-Lee, Hendler et al. 2001) provides many new perspectives and technologies to overcome the limitation of the WWW. Ontologies are a key component to solve the problem of semantic heterogeneity, and thus enable semantic interoperability between different web applications and services. Ontology is similar to a dictionary or glossary, but with greater detail and structure that enables computers to process its content. Ontology consists of a set of concepts, relations, and axioms that formulate a field of interest. Halaschek-Wiener et al. [3] mentioned several reasons why ontologies can help image retrieval. The first reason can be found in the fact that ontologies provide the ability to model the semantics of what occurs in images such as object, events, etc. The expressivity of the current Web ontology standard, OWL, allows for affiliated searches based on logic and structural inference. Ontologies also provide an elegant mechanism to formally organize image content in small, logically contained groups (ontological concepts), while enabling them to be linked, merged, and distinguished with other concepts in logically contained groups. Additionally, they enable the ability to assert that many images refer to the same concepts through the use of URIs. This, in turn allows these disparate information pieces to be linked together through image depictions. Consequently, the use of ontologies provides a new field with a lot of still open questions and really world-wide research. Because of all these advantages of the ontology and because of the lack of Arabic ontology which reflects negatively on the lack of systems which are based on the Arabic ontology, we selected this research topic to shed light on this important topic by putting some solutions through real e-work , examinations, and evaluations

Chapter Two

Background

2 Terms Definitions

2.1.1 Semantic Web we can say that the most frequent application of ontologies is in Semantic Web [5]. Semantic Web (SW) is an extension of very popular service of the Internet, World Wide Web (WWW). While WWW can be characterized as world-wide distributed web of linked documents using Uniform Resource Locators (URLs), SW is a web of linked data. In the former case, there are web-pages (documents) pointing to other web-pages. In the latter case, there are data items pointing to other data items by URLs. Due to this character of linked data, information about single entity can be distributed over the Web. Most of data model can still be accessible using SW technologies which it is part of SW infrastructure. It can only work due to common standards which data providers, middle-ware programmers and application developers uniformly use. Every year there is increasing number of fully-edged semantic applications within two main contests: Semantic Web Challenge and AI Mash up challenge.

2.1.2 Ontology [3] is a conceptualization of a domain into a human-understandable, but machine-readable format consisting of entities, attributes, relationships and axioms Ontology can provide a rich conceptualization of the working domain of an organization, representing the main concepts and relationships of the work

activities. These relationships could represent isolated information such as an employee's home phone number, or they could represent an activity such as a document, or attending a conference. The effective use of ontologies requires not only a well-designed and well defined ontology language, but also support from reasoning tools. Reasoning [6] is important both to ensure the quality of the ontology and in order to exploit the rich structure of ontologies and ontology based information. It can be employed in different phases of the ontology life-cycle. During ontology design, it can be used to test whether concepts are non-contradictory and to derive implied relations. In particular, one usually wants to compute the concept hierarchy, i.e. the partial ordering of named concepts based on the subsumption relationship. Information on which concept is a specialization of another, and which concepts are synonyms, can be used in designing test phase to make insure of concept definitions in the ontology.

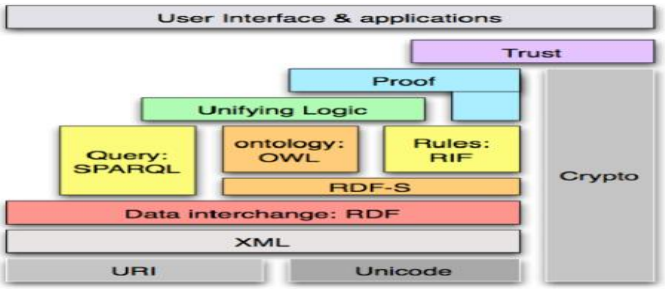


Figure 2.1.1: Layered Cake of SW Technologies and Standards [3].

2.1.3 RDFS [7] Web Ontological Schema Language Following W3C's 'one small step at a time' strategy, RDFS can be seen as a first try to support expressing simple ontologies with RDF syntax. In RDFS, predefined Web resources Class, Resource and Property can be used to define classes (concepts), resources and properties (roles).Unlike Dublin Core [20], RDFS does not predefine information

properties but a set of meta-properties that can be used to represent background assumptions in ontologies:

• RDFS: type: the instance-of relationship.

• RDFS: subclass Of: the property that models the subsumption hierarchy between classes.

• RDFS: subProperty Of: the property that models the subsumption hierarchy between properties.

RDFS: domain: the property that constrains all instances of a particular property, to describe instances of a particular class.

RDFS: range: the property that constrains all instances of a particular property, to have values that are instances of a particular class.

Table 2.1: RDFS ontology

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

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

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

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

xml:base=http://www.semanticweb.org/salydee/ontologies/2013/3/untitled-ontology-2

ontologyIRI=http://www.semanticweb.org/salydee/ontologies/2013/3/untitled-ontology-2 >

<Prefix name="" IRI="http://www.w3.org/2002/07/owl#"/>

<Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>

<Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>

<Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>

<Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>

<Class IRI="#تصنيف"/>

<ObjectAllValuesFrom>

<ObjectProperty IRI="#له_تصنيف"/>

```
<ObjectUnionOf>
<Class IRI="#ترفيهي"/>
<Class IRI="#حضاري"/>
<Class IRI="#ديني"/>
<Class IRI="#طبيعي"/>
```

2.1.4 RDF [7] is a directed, labeled graph data format for representing information in the Web. This specification defines the syntax and semantics of the SPARQL query language for RDF. SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries can be results sets or RDF graphs.

Table 2.1.4: SPARQL Example

```
PREFIX rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
PREFIX owl: http://www.w3.org/2002/07/owl#
PREFIX xsd: http://www.w3.org/2001/XMLSchema#
PREFIX rdfs: http://www.w3.org/2000/01/rdf-schema#
PREFIX Samah:http://www.semanticweb.org/salydee/ontologies/2013/4/untitled-ontology-4#
SELECT ?subject ?object
WHERE { ?subject rdfs:subClassOf  PalTousiom:تصنيف
OR ?subject rdfs:subClassOf  PalTousiom:تصنيف
```

2.1.5 Relational Search [8] Semantic recommending is related to relational search, where the idea is to try to search and discover serendipitous semantic associations between deferent content items. The idea is to make it possible for the end-user to

formulate queries such as “How is X related to Y” by selecting the end-point resources, and the search result is a set of semantic connection paths between X and Y.

2.1.6 Semantic Interoperability “the ability of information and communication technology systems and the business processes they support to exchange data and to enable the sharing of information and knowledge, also it enables systems to combine received information with other information resources and to process it in a meaningful manner. It aims at the mental representations that human beings have of the meaning of any given data” (Patel, Koch et al. 2004).

2.1.7 Semantic Gap [9] is described as; the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation. More precisely the gap means the difference between ambiguous formulation of contextual knowledge in a powerful language (e.g. natural language) and its sound, reproducible and computational representation in a formal language (e.g. programming language). Semantics of an object depends on the context it is regarded within. For practical application this means any formal representation of real world tasks requires the translation of the contextual expert knowledge of an application (high-level) into the elementary and reproducible operations of a computing machine (low-level). Since natural language allows the expression of tasks which are impossible to compute in a formal language there are no means to automate this translation in a general way.

2.1.8 A Morphological Analysis Technique is a computational process that analyzes natural words by considering their internal structures. The internal structure of a word may include stem, root, affixes, and patterns. Morphological

analysis techniques can be viewed as clustering mechanisms and usually help in resolving lexical ambiguity

2.1.9 Stemming [10] is a method of word standardization used to match some morphologically related words. The stemming algorithm is a computational process that gathers all words that share the same stem and have some semantic relation.

2.1.10 Image Retrieval System [11] is a computer system for browsing, searching and retrieving images from a large database of digital images. There are four basic approaches to image retrieval: the first: Content-based image retrieval depends on lowest level features such as color, texture, shape, and spatial location. The second: Text based image retrieval Image search is supported by augmenting images with keyword-based annotations and the search process always relies on keyword matching techniques. The third: Systems which are based on metadata properties like title, creator, resolution, image format, date and location can be considered. The fourth: the Semantic Web which provides new insights into the image retrieval problem, developing techniques to annotate the content of images by using ontologies.

2.2 Related Work

Many multimedia retrieval systems were developed in the 90s both for commercial and research purposes like QBIC [12], Virage [13], Swoogle[6], and many others like MARS(Huang et al, 1997|), Photobook (Pentland et al, 1996) or Excalibur (Wilf, 1998). Some years later the basic concept of similarity search was transferred to several Internet image search engines including Webseek [16] and Webseer [14].

It is important to mention the efforts made to integrate CBIR with enterprise databases such as Informix data blades, IBM DB2 Extenders, or Oracle Cartridges with the objective of making CBIR more accessible to the industry. Smeulders et al [15] give an exhaustive overview of the state of the art in CBIR before the year 2000. They identify three main categories based on user interaction: category search, target search and search by association.

The new direction was toward designing systems which would be user friendly and could bring the vast multimedia knowledge from libraries, databases, and collections to the world. To do this it was noted that the next evolution of systems would need to understand the semantics of a query, not simply the low level underlying computational features. This general problem was called "bridging the semantic gap''. In this section we review the 6 top-famous systems, the techniques that this six systems used in their approach are diverse and based on the state-of-art approaches. In reviewing these systems, we are reviewing the latest developments in this area.

2.2.1 Soo et al. [17] proposed a framework that can facilitate image retrieval based on a sharable domain ontology and thesaurus. They used case-based learning (CBL) which depends on a natural language phrase parser which proposed to convert a natural language query into resource description framework (RDF) format, a semantic-web standard of metadata description that supports machine readable semantic representation. This same parser also is extended to perform semantic annotation on the descriptive metadata of images and convert metadata automatically into the same RDF representation. The retrieval of images then can be conducted by matching the semantic and structural descriptions of the user query with those of the annotated descriptive metadata of images, they also developed two similarity comparison

algorithms -- MCS and MLRM -- over the NL phrases that can facilitate the retrieval mechanism for finding a most similar case from the case-based learning using a natural language query parser to translate a natural query into query in RDF format. The parser is able to perform semantic annotation on the descriptive metadata of images and convert metadata automatically into RDF representation. Their framework language was Chinese, the annotations. The collection used is a set of historical and cultural images that have been taken from Dr. Ching-chih Chens "First Emperor of China" defined and derived from a set of domain concepts. The ontology used is a Mandarin Chinese thesaurus.

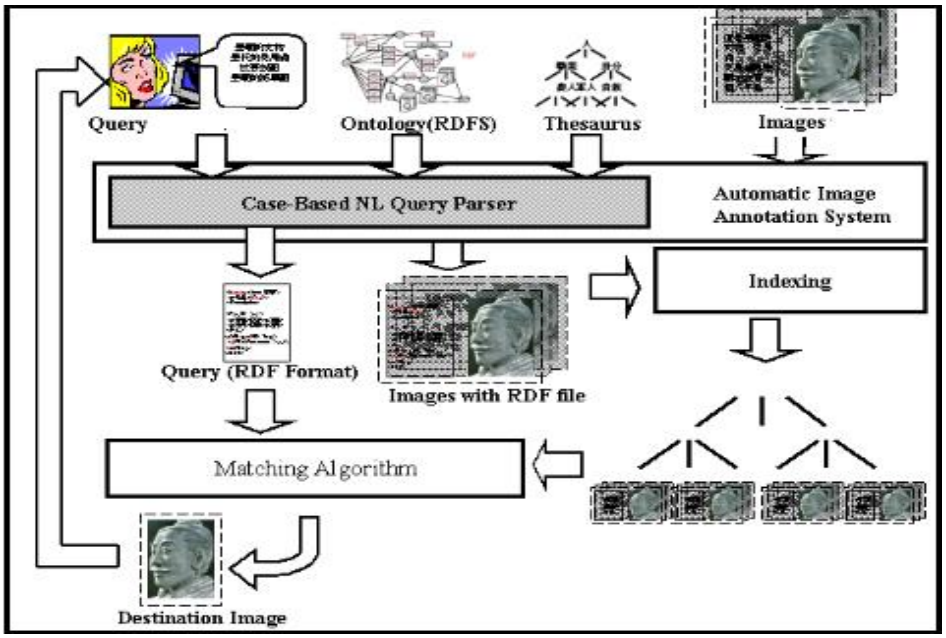


Figure 2.2.1: Soo Framework Diagram[17].

2.2.2 Heinecke [30], built transforming textual annotations of multimedia contents into an ontological representation (based on an existing ontology) in order to make them available for a knowledge-base; and translating English and French user queries into an ontological query language (SPARQL). The matching of linguistic data (lexicons, thesauri) with

ontologies is similar, but not identical to ontology matching or ontology alignment, trying to find corresponding classes of two ontologies. Different methods of matching were discussed; they used relationships found in the lexicon via a semantic thesaurus, and the taxonomic hierarchies of both.

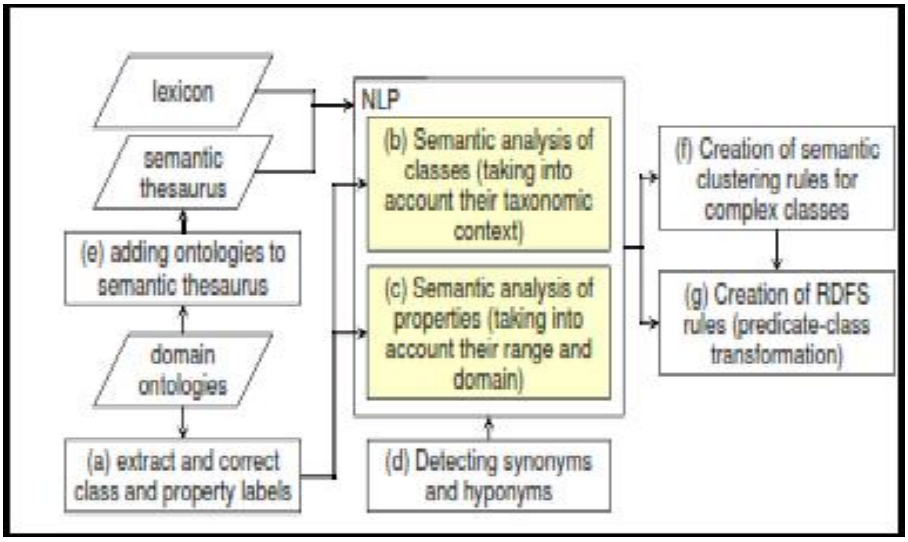


Figure 2.2.2: Heinecke Prototype [30].

2.2.3 Saathoff et al [18] presented an approach that combines multimedia reasoning and natural language processing for the semantic integration of automatic and manual image annotations based on domain ontologies. They discuss how to apply natural language processing to transform natural language descriptions and queries into an ontological representation that allows users to formulate formal semantics in an intuitive manner, without the need to cope with complex ontological structures and unwieldy user interfaces. Illustrative experimental examples demonstrate the added value, by using ontologies enriched with low-level features to label regions in images with semantic concepts.

2.2.4 Hare et al.[19] proposed the mechanism of generating automatically semantics for multimedia entities as bottom-up approach in opposition to the top-down approach that

consists of annotating images using ontologies, also proposed methods to use hierarchies induced on annotation words to improve automatic image annotation. While hierarchical clustering models have been explored for the image annotation problem, the hierarchies were statistically derived from image clusters. They presented a method for generating visual vocabularies based on the semantics of the annotation words and their hierarchical organization in ontology like WordNet. The semantically-motivated K-means clustering for generating the blobs improved the performance of the translation models for image annotation. They have been working with a variety of European cultural heritage institutions, including museums, galleries, and pictures. They considered that a combination of both approaches can lead to bridge the semantic-gap.

2.2.5 S. Petridis et al[20] They aimed to explicit knowledge representation which aims among others for supporting audio-visual content analysis and object-event recognition, the creation of knowledge beyond object and scene recognition through reasoning processes by enabling user friendly and intelligent search and retrieval. They presented knowledge infrastructure consists of several parts, namely the core ontology as a basis for all components of the knowledge infrastructure, multimedia-specific conceptualizations in form of the visual descriptor ontology and the multimedia structure ontology and a user-friendly visual descriptor extraction tool that allows the initialization of domain ontologies with visual features. Ontologies are extended and enriched through the Visual Descriptors Ontology and the Multimedia Structure Ontology to include low-level audiovisual descriptors in order to support automatic content annotation. Appropriate knowledge representation formalisms, core ontology and a user-friendly tool that allows the initialization of domain ontologies with visual features have been developed to complete this infrastructure.

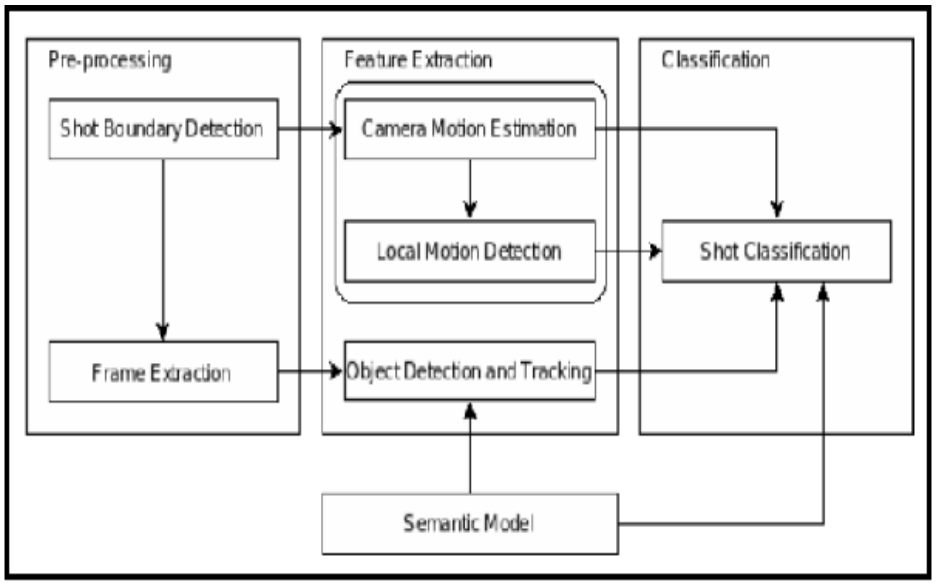


Figure 2.2.5: S. PETRIDIS Framework[20]

2.2.6 M.Srikanth et al. [21] machine learning approaches have been explored to model the association between words and images from an annotated set of images and generate annotations for a test image. They proposed methods to use a hierarchy defined on the annotation words derived from text ontology to improve automatic image annotation and retrieval. Specifically, the hierarchy is used in the context of generating a visual vocabulary for representing images and as a framework for the proposed hierarchical classification approach for automatic image annotation. The effect of using the hierarchy in generating the visual vocabulary is demonstrated by improvements in the annotation performance of translation models. In addition to performance improvements, hierarchical classification approaches yield well to constructing multimedia ontologies, using lexical resources like WordNet to induce hierarchies in the annotation words. They also proposed methods to use hierarchies induced on annotation words to improve automatic image annotation. While hierarchical clustering models have been explored for the image annotation problem, the hierarchies were statistically derived from image clusters. They presented a method

for generating visual vocabularies based on the semantics of the annotation words and their hierarchical organization in ontology like WordNet and the Corel collection of data.

2.3 Related Work In Arabic Ontology

To the best of our knowledge, there is no framework for images retrieval based on Arabic ontology, or an approved official Arabic ontology language approved yet.

There was a project undertaken by the CIA three years ago called (Arabic Word Net) which depends on translate (English word Net) to the corresponding words in Arabic, but the project did not succeed because the translation cannot provide a successful ontology, as the basic principle of ontology concepts based on the logical relationship between them [22].

Birzeit University start working in Arabic Ontology project recently, but it is not officially supported until now (http://sites.birzeit.edu/comp/ArabicOntology/) they developed the top levels of the Arabic Ontology, built manually based on DOLCE and SUMO upper level ontologies.

2.4 Comparison between our framework and previous related frameworks:

Like ours, all systems used multiple similarities, and thus faced the problem of aggregating similarities in an effective way. Therefore, reviewing the approaches helps us evaluate our approach, by benefitting from previous systems advantages and moving away in our development. There are many differences between our framework and the previous frameworks. First we built two prototypes for image retrieval system one based on ontology and the other based on relational data to compare between the two systems which more accurate, the first has its performance from database querying advantage like (light

search, fast search and accuracy), but second which depend on ontology, has its performance from relational concepts which make system more accurate and more efficient, results depend on comparing between two systems their based totally different, previous systems made matching between ontologies because they depend on convert all database to ontology languages, on one side, their quires is always specific and limited in certain words but we make our user query is opened to natural language of the user. Also, the all previous systems were designed to take into account the specific WordNet [23] internal forming its original language or by translate, but our framework is based on Arabic ontology lexically and linguistically, so our developing theory of semantic matching will be new theory to improve quality and efficiency of image retrieval based on Arabic ontology and produces high-quality results in terms of precision and recall indicators.

Chapter Three

Arabic Morphologic & Ontology

3.1 Arabic Particularity

Modern Standard Arabic [24] (MSA, العربية الفصحى الحديثة) is the official language of the Arab World. MSA is the primary language of the media and education. MSA is syntactically, morphologically and phonologically based on Classical Arabic (CA, العربية الفصحى التراثية), the language of the Qur'an (Islam's Holy Book). Lexically, however, MSA is much more modern. MSA is primarily written not spoken. The Arabic dialects, in contrast, are the true native language forms. They are generally restricted in use for informal daily communication. They are not taught in schools or even standardized although there is a rich popular dialect culture of folk-tales, songs, movies, and TV shows. Dialects are primarily spoken not written. However, this is quite changing as more Arabs are gaining access to electronic media of communication such as emails and newsgroups. Arabic dialects are loosely related to Classical Arabic. They are the result of the interaction between different ancient dialects of Classical Arabic and other languages that existed in neighbored and/or colonized what is today the Arab world [4]. For example, Algerian Arabic has a lot of influences from Berber as well as French.

Arabic dialects substantially differ from (MSA), and each other in terms of phonology, morphology, lexical choice and syntax.

3.2 Arabic Encoding, Input and Display

An encoding is a systematic representation of the symbols in a script for the purpose of consistent storage and access (data entry and display) by machines. The representational choices made in an encoding must be synchronized with data entry and display tools. The Arabic script brings certain challenges to the question of encoding design and how it interacts with data storage and access. This is primarily a result of how Arabic script is different from European scripts, whose handling has been the historical default. The basic challenges are the right-to-left directionality, contextually variant letter shapes, ligatures and the use of diacritics. In the extreme, an encoding can represent each complex ligature and letter shape with different diacritics as a separate complex "symbol." The number of different symbols in the encoding becomes very large. On the other extreme, different letter marks can be encoded as separate symbols from letter forms (and diacritics). Most commonly used encodings for Arabic, such as Unicode, CP-1256, ISO-8859 (among others), encode Arabic in logical ordered (first to last) graphemes of letters and diacritics. Basically, the fact that Arabic is displayed in a different direction on the screen from Roman script is considered irrelevant to the encoding as are the issues of contextual shaping and diacritization [24]. This encoding design choice makes Arabic storage efficient although it places the burden of correct display on the operating system or the specific program that displays Arabic.

3.3 Elements of the Arabic Script

The Arabic script is an alphabet written from right to left. There are two types of symbols in the Arabic script for writing words: letters and diacritics. In addition to these symbols, we discuss digits, punctuation and other symbols in this section.

3.3.1 Letters

Arabic letters are written in cursive style in both print and script (handwriting). They typically consist of two parts: letter form (رسم rasm) and letter mark (أعجام Ai&jAm). The letter form is an essential component in every letter. There is a total of 19 letter forms. The letter marks, also called consonantal diacritics, can be subclassified into three types. First are dots, also called points, of which there are five: one, two or three to go above the letter form and one or two to go below the letter form. Second is the short Kaf, which is used to mark specific letter shapes of the letter Kaf (see Figure 3.2). Third is the Hamza (همزةhamza~) letter mark [25]. The Hamza can appear above or below specific letter forms. The term Hamza is used for both the letter form (Z) and the letter mark, which appears with other letter forms such as أ Â, ؤ ˆw, and ئ ˆy. The Madda letter mark (مدةmad_a~) is a Hamza variant. [26]

`	w	r	d	A	l	k	h	T	S	s	q	f	m	γ	j	y	n	b	
ء	و	ر	د	ا	ل	ك	ه	ط	ص	س	ق	ف	م	غ	ج	ي	ن	ب	Isolated
					لـ	كـ	هـ	طـ	صـ	سـ	قـ	فـ	مـ	غـ	جـ	يـ	نـ	بـ	Initial
	ـو	ـر	ـد	ـا	ـل	ـك	ـه	ـط	ـص	ـس	ـق	ـف	ـم	ـغ	ـج	ـي	ـن	ـب	Medial
					ـل	ـك	ـه	ـط	ـص	ـس	ـق	ـف	ـم	ـغ	ـج	ـي	ـن	ـب	Final

Figure 3.3.1: A sample of letters with their deferent letter shapes [25].

3.3.2 Diacritics

The second class of symbols in the Arabic script is the diacritics. Whereas letters are always written, diacritics are optional: written Arabic can be fully diacritized, partially diacritized, or entirely undiacritized. The NLP task of restoring diacritics, or simply diacritization. Typically, Arabic text is undiacritized except in religious texts and children educational texts. Some diacritics are indicated in modern written Arabic to help readers

disambiguate certain words. In the Penn Arabic Treebank [26], 1.6% of all words have at least one diacritic indicated by their author. Out of these, 99.3% are actually correct, as in they appear in the correct position in the word.

3.3.3 Arabic Encodings

Many different "standard" encodings were developed for Arabic over the years. We only discuss here the three most commonly used encodings, which are all well supported for input and output on different platforms. Tables 2.1 and 2.2 present the different code values used for MSA Arabic symbols in multiple encodings side-by-side 8-bit Encodings: ISO-8859-6 and CP-1256 ISO-8859-6 and CP-1256 are two of the most popular early encoding schemes of Arabic.ISO-8859 was developed by the International Standards Organization. CP-1256, aka Arabic Windows encoding, was developed by Microsoft and made extremely popular through Windows. Both of these encodings use 1 byte (8-bits) to represent every single symbol (maximum of 256 characters). As in other encodings in their class for scripts/languages other than Arabic, the first 7-bits (or 128 characters) are reserved for English ASCII (American Standard Code for Information Interchange). The other script is represented in the other 128 characters. This allows the same encoding to be used for two scripts (or multiple languages) if needed. In CP-1256, the Arabic characters are listed in order although with some gaps in between different sets of characters to allow for maintaining the code values for some European languages, particularly French, thus effectively producing a multilingual code page (Arabic, English, French) that can be used in both Anglophone and Francophone Arab countries. Both CP-1256 and ISO-8859-6 couldn't accommodate the full set of extended Arabic characters [28]; however, characters from Persian are included. These encodings specify the graphemes only and rely on separate algorithms to display the correct font glyphs. CP-1256 and ISO-8859 are not compatible although they agree on the first 22 characters. This simple fact

means that words made up completely from characters in this overlapping set will look "correct" in either encoding. For example, see the word(حرة) When verifying the encoding of a sorted list of words (as in a dictionary), it is wise to look beyond the first few words to avoid falling for this ambiguity.

3.3.4 Unicode

Unicode is the current de facto standard for encoding a large number of languages and Scripts simultaneously. Unicode originally was designed to use two bytes of information (to code 65,536 unique symbols) and has been expanded since to cover over 1 million unique symbols. For Arabic, Unicode supports an extended Arabic character set. It also gives Arabic letter shapes and ligatures unique addresses under what it calls Presentation Forms A and B charts. Because Unicode encodes so many more characters than ISO8859-6 and CP-1256[28], conversion from these encodings into Unicode is possible, but the reverse may be lossy.

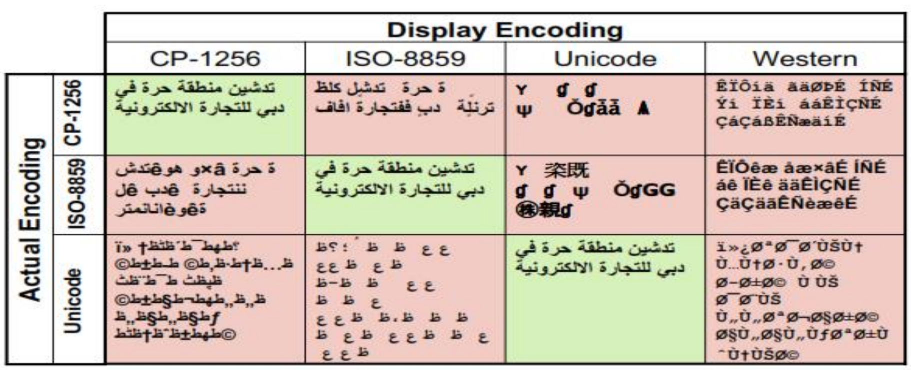


Figure 3.3.4: Comparing the correct and incorrect decoding of various Arabic encodings [28].

3.4 Arabic Treebank's

Collections of manually checked syntactic analyses of sentences, or Treebank's, are an important resource for building statistical parses and evaluating parsers in general. Rich Treebank annotations have also been used for a variety of applications such as tokenization, diacritization, part-of-speech (POS) tagging, morphological disambiguation, base phrase chunking, and semantic role labeling. Under time restrictions, the creation of a Treebank faces a trade off between linguistic richness and Treebank size. This is especially the case for morph-syntactically complex languages such as Arabic or Czech . Linguistically rich representations provide many (all) linguistic features that may be useful for a variety of applications. This comes at the cost of slower annotation as a result of longer guidelines and more intense annotator training. As a result; the richer the annotation, the slower the annotation process and, the smaller the size of the Treebank. Consequently, there is less data to train tools. In the case of Arabic, two important rich-annotation Tree banking efforts exist: the Penn Arabic Treebank (PATB) [26]. and the Prague Arabic Dependency Treebank(PADT) [63]. Both of these efforts employ complex and very rich linguistic representations that require a lot of human training. The amount of details specified in the representations is impressive. The PATB not only provides tokenization, complex POS tags, and syntactic structure; it also provides empty categories, diacritizations, lemma choices and some semantic tags. This information allows for important research in general NLP applications; however, much of this rich annotation is currently unused in Arabic parsing research [27] since it is generally considered to be derivative of the output of parsing itself. For example, nominal case: which can be determined for gold syntactic analyses at high accuracy, cannot be predicted well in a pre-parsing POS tagging step [28, 27]. To address this issue, a third Treebank, the Columbia

Arabic Treebank (CATIB), was recently introduced with the goal of speeding up annotation through representation simplification [63].

3.4.1 The Penn Arabic Treebank

The Penn Arabic Treebank (PATB) project started in 2001 at the Linguistic Data Consortium (LDC) and the University of Pennsylvania, the birthplace of Treebank's for English, Chinese, and Korean three parts of the PATB have been released publicly through the LDC (almost 650K words) and four other parts, including a Levantine Arabic Treebank [26], have been developed for the DARPA-funded projects. Each PATB part was released in different versions with different degrees of improvements. The PATB is annotated for morphological information, part-of-speech, English gloss, and for syntactic structure in the phrase-structure style of the Penn (English) Treebank (PTB) .The PTB guidelines are modified to handle Arabic. For example, Arabic verbal subjects are analyzed as verb phrase (VP) internal; following the verb. The creation of the PATB is a great achievement for Arabic NLP. This resource has been crucial for so much research in morphological analysis, disambiguation, POS tagging and tokenization, not to mention of course parsing. Every other Treebank created since PATB, has used it or some of the tools developed for it. For instance, both the Prague dependency Treebank and the Columbia Arabic Treebank converted the PATB to their own representation in addition to annotating additional data. Another example of the importance of the PATB is that its tokenization is the de facto standard for most Arabic Tree banking efforts.

3.4.2 The Prague Arabic Dependency Treebank

The Prague Arabic Dependency Treebank (PADT) is maintained by the Institute of formal and applied Linguistics, Charles University in Prague. PADT contains a multilevel description comprising functional morphology, analytical dependency syntax, and grammatical representation of linguistic meaning. These linguistic annotations are based on the Functional Generative Description theory [27] and the Prague Dependency Treebank project [28].

3.4.3 Columbia Arabic Treebank

The Columbia Arabic Tree Bank (CATIB) project was started in Columbia University in 2008. It contrasts with previous Arabic Tree banking approaches in putting an emphasis on faster production with some constraints on linguistic richness [63]. Two ideas inspire the CATIB approach. First, CATIB avoids annotation of redundant linguistic information. For example, nominal case and state (definite, indefinite, construct) in Arabic are determined automatically from syntax and morphological analysis of the words and need not be annotated by humans. Of course, some information in CATIB is not easily recoverable, such as phrasal co-indexation and full lemma disambiguation. Second, CATIB uses a linguistic representation and terminology inspired the long tradition of Arabic syntactic studies. This makes it easier to train annotators, who need not have degrees in linguistics. CATIB uses an intuitive dependency structure representation and relational labels inspired by Arabic grammar such as (tamyiz) and (idafa) in addition to the well-recognized labels of subject, object and modifier.

Table 3.4: Comparison between PATB, PADT and CATIB

The Criteria	PATB	PADT	CATIB
Syntactic Structure.	annotate heads explicitly and spans of phrases/clauses implicitly	annotate heads explicitly and spans of phrases/clauses implicitly	annotate heads explicitly and spans of phrases/clauses implicitly
Syntactic and Semantic Functions	uses about 20 dashtags that are used for marking syntactic and semantic functions. Syntactic dash tags include -TPC and -OBJ	uses around 20 labels, although with different functionality from PATB and CATiB. In general, PADT analytical labels are deeper than CATiB since they are intended to be a stepping stone towards the PADT tec to grammatical level.	CATiB's relation labels mark syntactic function only. The use of the syntactic labels SBJ and TPC is different between CATiB and PATB. In PATB, TPC is used to mark the subject or object when they appear before the verb. Further co-indexation is used to specify the role of the TPC inside the verb phrase.
Empty Pronouns	annotated in PATB PADT nor CATiB. Verbs with no explicit subjects in CATiB (and PADT) can be assumed to pro-drop	Not annotated	Empty Pronouns. Empty pronouns are annotated in PATB but not PADT nor CATiB. Verbs with no explicit subjects in CATiB (and PADT) can be assumed to pro-drop
Coreference.	Coreference indices are annotated in PATB for	PADT only annotates coreference between	CATiB does not annotate any

	traces and explicit pronouns. .	explicit pronouns and what they corefer with	coreference indices.
Word Morphology.	uses over 400 tags specifying every aspect of Arabic word morphology such as definiteness, gender, number, person, mood, voice and case.	As for parts-of-speech, PATB PADT morphology is more complex than PATB. For instance, it makes more sophism.	uses the same basic tokenization scheme used by

3.5 Related Arabic Parsers

There are a number of computational implementations for parsing Arabic. Daimi 2001[27] developed a syntactic parser for Arabic using the Definite Clause Grammar formalism. Žabokrtský and Smrž 2003, developed a dependency grammar for Arabic, with a focus on the automatic transformation of phrase-structure syntactic trees of Arabic into dependency-driven analytical ones. A probabilistic parser for Arabic is being developed at the Dublin City University based on the Arabic Penn Treebank Corpus (Al-Raheb et al., 2006). The Stanford Natural Language Processing Group has developed an Arabic parser based on PCFG (Probabilistic Context-Free Grammar) using the Penn Arabic Treebank. Othman et al. (2003) developed a chart parser for analyzing Arabic sentences using Unification-based Grammar formalisms. Ramsay and Mansour (2007) wrote a grammar for Arabic within a general HPSG-like framework (Head-driven Phrase Structure Grammar) for the purpose of constructing a text-to-speech system. Within the framework

of corpus linguistics, Ditters (2001) wrote a grammar for Arabic using the AGFL formalism (Affix Grammars over a Finite Lattice) There are mainly two strategies for the development of Arabic morphologies depending on the level of analysis:

1. Stem-based morphologies: analyzing Arabic at the stem level and using regular concatenation. A stem is the least marked form of a word, which is the uninflected word without suffixes, prefixes, proclitics or enclitics. In Arabic, this is usually the perfective, person, singular verb, and in the case of nouns and adjectives they are in the singular indefinite form.

2. Root-based morphologies: analyzing Arabic words as composed of roots and patterns in addition to concatenations. A root is a sequence of three (rarely two or four) consonants which are called radicals, and the pattern is a template of vowels, or a combination of consonants and vowels, with slots into which the radicals of the root are inserted ,this process of insertion is usually called interdigitation (Beesley, 2001) [35].

3.6. Our Parser Methodology:

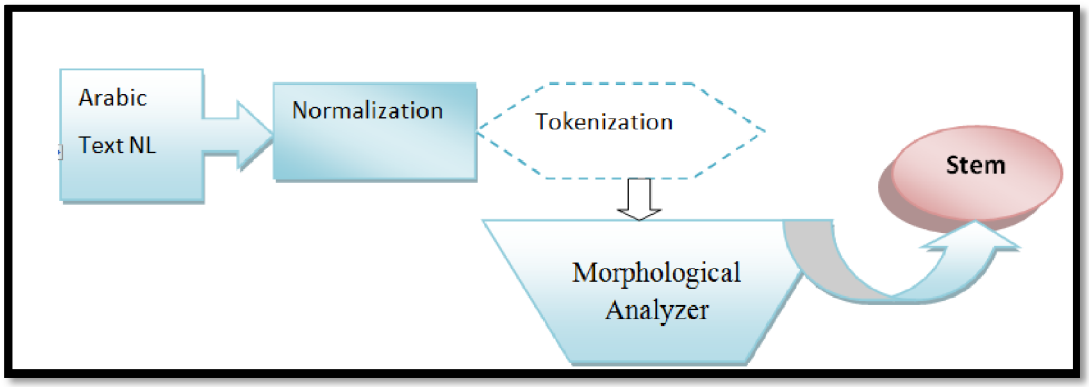


Figure 3.6: Our parser Methodology

3.6.1 Normalization

Normalization is a preliminary stage to tokenization where preliminary processing is carried out to ensure that the text is consistent and predictable. In this stage, for example, the decorative elongation character, kashida, and all diacritics are removed. Redundant and misplaced white spaces are also corrected, to enable the tokenizer to work on a clean and predictable text. In real-life data spaces may not be as regularly and consistently used as expected. There may be two or more spaces, or even tabs, instead of a single space. Spaces might even be added before or after punctuation marks in the wrong manner. Therefore, there is a need for a tool that eliminates inconsistency in using white spaces, so that when the text is fed into a tokenizer or morphological analyzer, words and expressions can be correctly identified and analyzed.

3.6.2 Tokenization

It is a necessary and non-trivial step in natural language processing. The function of a tokenizer is to split a running text into tokens, so that they can be fed into a morphological transducer or POS tagger for further processing. The tokenizer is responsible for defining word boundaries, demarcating clitics, multiword expressions, abbreviations and numbers. A token is the minimal syntactic unit; it can be a word, a part of a word (or a clitic), a multiword expression, or a punctuation mark. A tokenizer needs to know a list of all word boundaries, such as white spaces and punctuation marks, and also information about the token boundaries inside words when a word is composed of a stem and clitics. Throughout this research full form words, i.e. stems with or without clitics, as well as numbers will be termed main tokens. All main tokens are delimited either by a white space or a punctuation mark. Full form words can then be divided into sub-tokens, where clitics and stems are separated.

3.6.3 Morphological Analysis

A morpheme is the smallest element that has a meaning. Some morphemes exist as words at the same time. Morphemes cannot be split into smaller ones, and they should impart a function or a meaning to the word which they are part of (Spencer, 1991). The root is a single morpheme that provides the basic meaning of a word (Spencer, 1991). Generally speaking, in English, the root is sometimes called the word base or stem; it is the part of the word that remains after the removal of affixes (Al-Khuli, 1991). In Arabic, however, the base or stem is different from the root (Al-Atram, 1990). In Arabic the root is the original form of the word before any transformation process, and it plays an important role in language studies (Metri & George, 1990). Defective or weak roots are the roots with one or more long vowels. A stem is a morpheme or a set of concatenated morphemes that can accept an affix (Al-Khuli, 1991). The stem expresses some central idea or meaning (Paice, 1994). An affix is a morpheme that can be added before or after, or inserted inside, a root or a stem as a prefix, suffix or infix, respectively; to form new words or meanings (Al-Khuli, 1991; Thalouth & Al-Dannan, 1987). Arabic prefixes are sets of letters and articles attached to the beginning of the lexical word and written as part of it, while suffixes are sets of letters, articles, and pronouns attached to the end of the word and written as part of it (Al-Atram, 1990). English has 75 prefixes and about 250 suffixes (Salton, 1989). Arabic has fewer affixes. Arabic affixes have the feature of concatenating with each other in predefined linguistic rules. This feature increases the overall number of affixes (Ali, 1988). The removal of prefixes in English is usually harmful because it can reverse or otherwise alter the meaning or grammatical function of the word. This is not so in Arabic, since the removal of prefixes does not usually reverse the meaning of words. Word morphology usually refers to the different forms of individual words. These forms

express the type and function of the individual words. Word morphology is very helpful in the process of learning to use a dictionary efficiently and acquiring linguistic information. It also has an important role to play in the disambiguation of word sense. A morphological analysis technique is a computational process that analyzes natural words by considering their internal structures. The internal structure of a word may include stem, root, affixes, and patterns. Morphological analysis techniques can be viewed as clustering mechanisms and usually help in resolving lexical ambiguity. Clustering is a very useful process in many natural language applications (Krovetz, 1993) such as information retrieval (ElAffendi, 1998), text classification, and text compression. In the case of Arabic, the main purpose of any morphological analysis technique is to obtain the root of a given word (El-Affendi, 1991). This is true of many applications, but not of all. Stemming is a method of word standardization used to match some morphologically related words. The stemming algorithm is a computational process that gathers all words that share the same stem and have some semantic relation. The main objective of the stemming process is to remove all possible affixes and thus reduce the word to its stem.

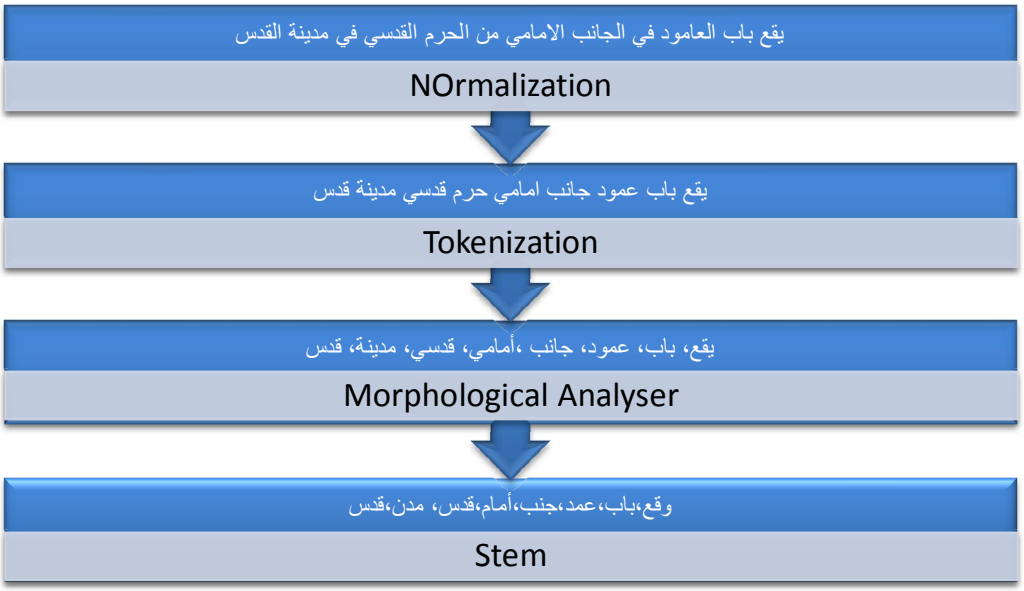


Figure 3.6.3: Arabic Stemmer System Example

Our methodology to build the stemmer including these steps:

1. Split the paragraph to tokens according to the spaces between the words.
2. Remove all stop words, and all words under three letters.
3. Remove the prefix and the suffixes; we will stop if what is left of the word contains a valid stem.
4. Once a stem is found, a pattern is then constructed by replacing the root letters from the remaining part of the word with the letters of the basic pattern.

   For lists of prefixes, suffixes, verb roots, solid word roots, patterns, foreign words, and function words were created using statistical studies(we used shereen khoja Lists).

3.7. Ontology Definition

Ontology in general, is a shared understanding (i.e. semantics) of a certain domain, axiomatized and represented formally in a computer resource. By sharing ontology, autonomous and distributed applications can meaningfully communicate to exchange data and make transactions interoperate independently of their internal technologies.

The meaning in an ontology (i.e. the semantics of the vocabulary used in this ontology) is supposed to be represented in a logical form. In other words, an ontology becomes a logical theory where its logical statements (i.e. axioms) the intended meaning of the vocabulary [32]. ontologies in Artificial Intelligence (AI), the meaning of this concept still generates a lot of controversy, both within and outside of AI. The classical AI definition; Ontology is a formal specification of a conceptualization [30] that is; an abstract and simplified view of the world that we wish to represent, described in a language that is equipped with a formal semantics. In knowledge representation, ontology [31] is a description of the concepts and relationships in an application domain. Depending on the

users of this ontology, such a description must be understandable by humans or by software agents. In many other field – such as in information systems and databases, and in software engineering – ontology would be called a conceptual schema. An ontology is formal, since its understanding should be non ambiguous, both from the syntactic and the semantic point of views. Researchers in AI were the first developer of ontologies with the purpose of facilitating automated knowledge sharing.

Since the beginning of the 90's, ontologies have become a popular research topic, and several AI research communities, including knowledge engineering, knowledge acquisition, natural language processing, and knowledge representation, have investigated them. More recently, the notion of ontology is becoming widespread in fields such as intelligent information integration, cooperative information systems, information retrieval, digital libraries, e-commerce, and knowledge management. Ontologies are widely regarded as one of the foundational technologies for the Semantic Web: when annotating web documents with machine-interpretable information concerning their content, the meaning of the terms used in such an annotation should be fixed in a (shared) ontology. Research in the Semantic Web has led to the standardization of specific web ontology languages. An ontology language is a mean to specify at an abstract level – that is, at a conceptual level – what is necessarily true in the domain of interest. More precisely, we can say that an ontology language should be able to express constraints, which declare what should necessarily hold in any possible concrete instantiation of the domain [32]

3.7.1 What is ontology used for ?

- Sharing common understanding of the structure of information among people or software agents
- Enabling reuse of domain knowledge

- Making domain assumptions explicit
- Separating domain knowledge from the operational knowledge
- Analyzing domain knowledge

Normally the creation and design of ontology is not the main goal in itself. Ontology is, very roughly, a formal representation of a domain of knowledge. It is an abstract entity: it defines the vocabulary for a domain and the relations between concepts, but an ontology says nothing about how that knowledge is stored (as physical file, in a database, or in some other form), or indeed how the knowledge can be accessed [31]

A knowledge base is a physical artifact: it is a database, a repository of information that can be accessed and manipulated in some predefined fashion. The knowledge in a knowledge base can be said to be modeled according to Ontology [32]

3.8. Challenges of building Arabic Ontology

Our preliminary researching in the field of the SW revealed a number of issues that suggest reasons for the lack of Arabic research, namely the lack of technology support and adequate resources.

1) Lack of Arabic support in existing Semantic Web tools: a specific problem with SW tools processing Arabic text concerns encoding. Different encoding of Arabic script exists on the Web; dominant encodings include UTF-8, Windows-1256 and ISO-8859-6, most of the SW tools were built using Java, which supports internationalization. Therefore, there is a strong need to consolidate the different Arabic encoding or simply adhere to one encoding schema when representing Arabic text (Unicode). Typical SW developers' tools use Unicode throughout (Carroll, 2005); hence this might solve part of the support problem.

2) Lack of Arabic Semantic Web applications: another evidence of the lack of Arabic in the Semantic Web world is a recent statistic provided by the (OntoSelect) ontology library, which shows that 49% of the ontologies in the library are created in English. This problem could be attributed to the lack of tools and software development environments that process Arabic script in all steps of the semantic annotation process.

3) Limited support for Arabic research in the field of Semantic Web technologies

Most of the SW research is a result of investment from both grant bodies and academic research centers. SW tools, such as Protégé, GATE, and Jena to name just a few are all products of successful investment in the SW field. For the particular case of Arabic, the limited research problem can be attributed to the lack of adequate resources in terms of skills, funding and interest in this emerging field of Web research. The allocation of research funding, the provision of resources, and interest from a committed practice community are essential if we are to overcome this problem.

In terms of financial resources, a major concern -particularly for communities with a small user base such as Arabic users- is the cost of SW application development and maintenance. In some well structured areas such as scientific, commercial and government applications, the potential benefits in conductivity gain and profit will outweigh the cost of developing and maintaining an Ontology application. The cost, in terms of time and effort required, will decrease as the user base increases [28].

3.9. Building Ontology

There are six parts [29] in the means of creating the life cycle of ontology which are the following: Ontology creation, Ontology population, Ontology validation, Ontology deployment, Ontology maintenance and Ontology evolution .The ontology learning process into six steps can also be subdivided into extract term, discover synonyms, obtain

concepts, extract concept hierarchies, and define relations among concepts, deduce rules or axioms. These processes are used in order to make the ontology matching become possible and that the related branches of topics would be available to any users. It is not possible to build a logical theory to specify the complete and exact intended meaning of a domain vocabulary. In practice, the level of detail that is appropriate to explicitly capture and represent is subject to what is reasonable and plausible for applications. Other details will have to remain implicit assumptions. These assumptions are usually denoted in linguistic terms that we use to lexicalize concepts, and this implicit character follows from our interpretation of these linguistic terms, on the relationship between concepts and their linguistic terms

3.10. Tools for Creating Ontology

Table 3.10: Comparison between Ontology Tools in supporting Arabic

Tool	Creator	Functionality	Standards	Arabic Support
Protégé	Stanford Center for Biomedical Informatics Research	Ontology editor and knowledge base framework for ontology manipulation & query	RDF	Support
			RDFS	Support
			OWL	Limited Support
			SPARQL	Limited Support
Jena	Hewlett-Packard Development Company	Framework for ontology manipulation and query	RDF	Support
			RDFS	Support
			OWL	Support
			SPARQL	Limited Support
Sesame	Aduna in cooperation with NLnet Foundation	Framework for storage, inference and querying of RDF data	RDF	Limited Support
			RDFS	Limited Support
			OWL	Limited Support
			SPARQL	NO Support
KAON2	Suite of Research Center for Information Technologies	Ontology management (Create, Manipulate, Infer) tools	RDF	NO Support
			RDFS	NO Support
			OWL	NO Support

3.11. Design Goals

The design of RDF is intended to meet the following goals:

- Having a simple data model.
- Having formal semantics and provable inference.
- Using an extensible URI-based vocabulary.
- Using an XML-based syntax.
- Supporting use of XML schema data types.
- Allowing anyone to make statements about any resource.
- RDF has a simple data model that is easy for applications to process and manipulate.
- RDF is an open-world framework that allows anyone to make statements about any resource.
- Global view to heterogeneous, distributed contents. The contents of different content providers can accessed through one service as a single, seamless, and homogenous repository [7]
- Automatic content aggregation. Satisfying an end-user's information need often requires aggregation of content from several information providers, a task suitable for semantic web technologies. For example, when looking for data about an artist, relevant information may be provided by museum collections, libraries, archives, authority records, ontologies, and other sources.

Table 3.11: Part of our domain ontology

```
<Class IRI="#إسلامي"/>
</Declaration>
<Declaration>
<Class IRI="#احتلال"/>
</Declaration>
<Declaration>
<Class IRI="#اساسي"/>
</Declaration>
<Declaration>
<Class IRI="#الخط_الاخضر"/>
</Declaration>
<Declaration>
<Class IRI="#الضفة_الغربية"/>
</Declaration>
<Declaration>
<Class IRI="#العصر_الحديث"/>
</Declaration>
<Declaration>
<Class IRI="#العصر_العباسي"/>
</Declaration>
<Declaration>
<Class IRI="#العصر الأموي"/>
</Declaration>
<Declaration>
<Class IRI="#العصر الفاطمي"/>
</Declaration>
<Declaration>
<Class IRI="#العنوان"/>
</Declaration>
<Declaration>
<Class IRI="#الموقع"/>
</Declaration>
<Declaration>
<Class IRI="#انشاء"/>
</Declaration>
<Declaration>
<Class IRI="#بعد الاحتلال"/>
</Declaration>
<Declaration>
<Class IRI="#بلدة"/>
</Declaration>
<Declaration>
<Class IRI="#ترفيهي"/>
</Declaration>
<Declaration>
<Class IRI="#ترميم"/>
</Declaration>
<Declaration>
<Class IRI="#تصنيف"/>
</Declaration>
<Declaration>
<Class IRI="#ثانوي"/>
</Declaration>
<Declaration>
<Class IRI="#جزء"/>
</Declaration>
<Declaration>
<Class IRI="#حدث"/>
```

```
</Declaration>
<Declaration>
<Class IRI="#حرق"/>
</Declaration>
<Declaration>
<Class IRI="#حضاري"/>
</Declaration>
<Declaration>
<Class IRI="#ديني"/>
</Declaration>
<Declaration>
<Class IRI="#صورة"/>
</Declaration>
<Declaration>
<Class IRI="#طبيعي"/>
</Declaration>
<Declaration>
<Class IRI="#عصر_الخلفاء الراشدين"/>
</Declaration>
<Declaration>
<Class IRI="#عصر البهائيين"/>
</Declaration>
<Declaration>
<Class IRI="#عصر الرسول"/>
</Declaration>
<Declaration>
<Class IRI="#عصر الصعاليك"/>
</Declaration>
<Declaration>
<Class IRI="#غزة"/>
</Declaration>
<Declaration>
<Class IRI="#قرية"/>
</Declaration>
<Declaration>
<Class IRI="#كيان"/>
</Declaration>
<Declaration>
<Class IRI="#ما بعد الميلاد"/>
</Declaration>
<Declaration>
<Class IRI="#ما قبل الميلاد"/
```

Chapter Four

Comparison between the Database and the Ontology

Relational databases have been designed to store high volumes of data and to provide an efficient query interface. Ontologies are geared towards capturing domain knowledge, annotations.

Mainly Ontologies proliferate with the growth of the Semantic Web, in spite of most of data on the Web are still stored in relational databases.

4.1 Schemas Differences:

4.1.1 Database Schema

The relational database have many advantages depending on its schema, which made it efficient in computer systems such :

- Defines the structure of a database in a formal language.
- Refers to any of: conceptual, logical, physical loosely.
- Expresses for types, properties, constraints deeply.
- puts constraints for consistency checking.
- Shares meaning of some subject matter.
- Gives taxonomy.
- Used for multi-purpose.
- Considered as embedded natural language definitions.

- Gives constraints for meaning.
- Gives Constraints for ensuring
- Gives self-consistency(not data).
- Used as abstract types w/no instances.
- Reused to build new ones.
- Reused in unexpected ways.
- Gives formal model-theoretic semantics.

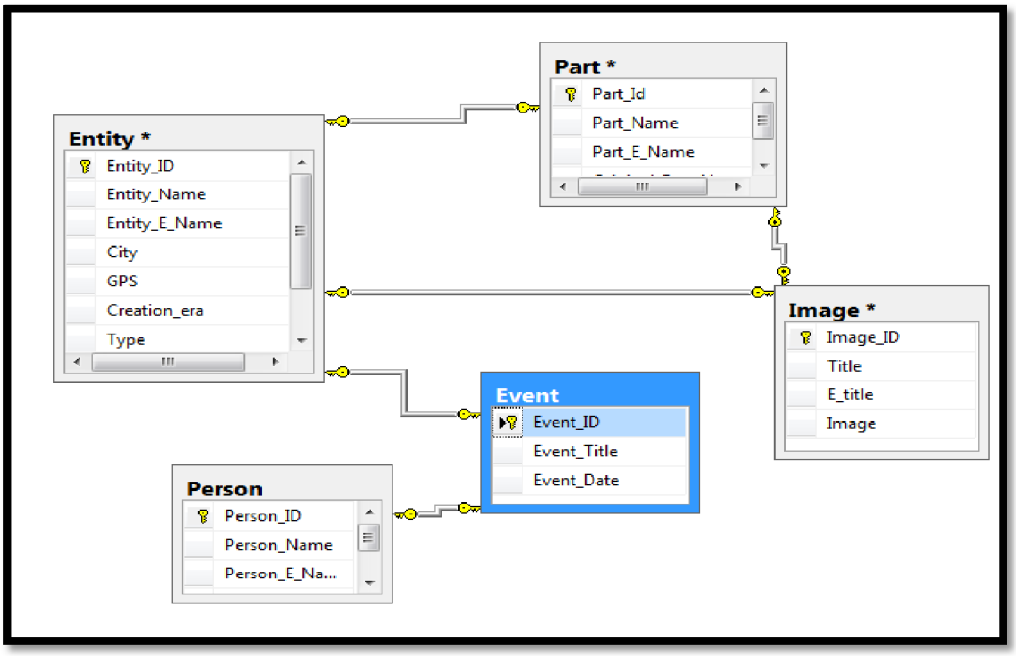


Figure 4.5.1: PalTourism Domain Database Schema

4.1.2 Ontology Schema

Ontology is comprised of four main components: concepts, instances, relations and axioms. The present research adopts the following definitions of these ontological components [32]:

- **A Concept** (also known as a class or a term) is an abstract group, set or collection of objects. It is the fundamental element of the domain and usually represents a group or class whose members share common properties. This component is

represented in hierarchical graphs, such that it looks similar to object-oriented systems. The concept is represented by a super-class, representing the higher class or so-called parent class, and a subclass which represents the subordinate or so-called child class. For instance, a university could be represented as a class with many subclasses, such as faculties, libraries and employees.

- **An Instance** (also known as an individual) is the ground-level component of an ontology, which represents a specific object or element of a concept or class. For example, "Jordan" could be an instance of the class "Arab countries".
- **A Relation** (also known as a slot) is used to express relationships between two concepts in a given domain. More specifically, it describes the relationship between the first concept, represented in the domain, and the second, represented in the range. For example, "study" could be represented as a relationship between the concept "person" (which is a concept in the domain) and "university" (which is a concept in the range).
    - **An Axiom** is used to impose constraints on the values of classes or instances, so axioms are generally expressed using logic-based languages such as first order logic; they are used to verify the consistency of the ontology.

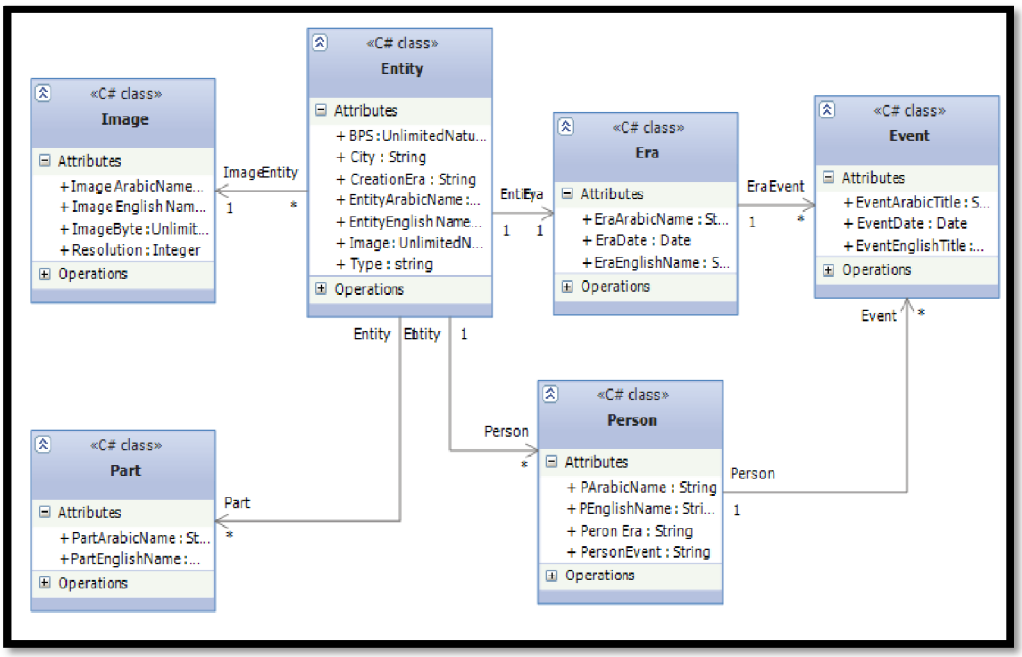


Figure 4.5.2: PalTourism Domain Ontology Schema

Table 4.1: The Main Differences between Database And Ontology

The function	DB Schema	Ontology
Focus	Data	Meaning Shared Understanding
Core Purpose(s)	Structure instances for efficient storage and querying	Human communication, interoperability, search, software engineering,
Notation: syntax	ER diagrams; no standard serialization syntax.	Logic; no standard diagram notation syntax.
Notation: semantics	Minimal focus on formal semantics.	Strong focus on formal semantics.
Expressivity overlap	Entities Attributes, Relations Constraints	Classes Properties Axioms
Expressivity differences	No Taxonomy Constraints for integrity,	Taxonomy is backbone. Constraints for

	foreign key, delete.	meaning, consistency & integrity.
Starting point	Scratch, rarely reuse.	Reuse if possible.
Normalization	Standard rules in natural language, little tool support.	No standard rules or guidelines.
Optimization	Fundamental step. Manual, geared to specific queries for specific DB	Ontology independent; Inference engine developers.
Change Management, Agility, Flexibility	Locked into specific set of queries per DB. Tight coupling. Lost meaning. Hard to evolve and maintain. ETL tools to help.	No query lock-in. Queries usable on other systems. Looser coupling. Semantics explicit. Potentially easier to evolve & maintain. Looser coupling. Semantics explicit. Potentially easier to evolve & maintain.
Processing Engines	SQL Engines Queries Reasoning with Views Data integrity Standardized on SQL	Theorem Provers Derive new information from existing information. Consistency and Integrity Less standardization
Performance	Highly tuned for performance and scale. Not work well with too many joins.	Full inferencing: much smaller scale. Reduced inferencing: reaching large scale

4.2. RDF Query Language (SPARQL)

An RDF query language is a query language for databases, able to retrieve and manipulate data stored in Resource Description Framework format. It was made a standard by the RDF Data Access Working Group (DAWG) of the World Wide Web

Consortium, and is recognized as one of the key technologies of the semantic web. On 15 January 2008, SPARQL 1.0 became an official W3C recommendation and SPARQL 1.1 in March, 2013[7].

Table 4.4.1: A SPARQL query to get the addresses for each tourism place in(القدس)

```
SELECT ? Title?city
WHERE {
    ?who <Title#E_name> ?Entity ;
    <Title#addr> ?adr .
    ?adr <Address#city> ?city ;
    <Address#state> "القدس"
```

4.3 Relational Database Query Language (SQL)

SQL stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system. However, the standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database.

Table 4.4.1: A SQL query to get the addresses for each tourism place in(القدس)

```
SELECT Title, Address
FROM Entity, Address
WHERE Title.addr=Address.ID
AND Address.state="القدس "
```

4.4. Differences between SPARQL and SQL

Both of these languages give the user access to create, combine, and consume structured data. SQL does this by accessing tables in relational databases, and SPARQL does this by accessing a web of Linked Data. (Of course, SPARQL can be used to access relational data as well, but it was designed to merge disparate sources of data.)

Chapter Five

Implemantaion

We design two prototypes (PalTourism) that can retrieve tourism Images one using sharable domain ontology mapping, and the other by using relational database.

5.1 The First Prototype (Ontology Based):

The Semantic Web data development life cycle follows these steps:

1. Storage: The framework must acquire or reference existing space, typically in memory or a database, to store Semantic Web data.

2. Population: The framework populates the referenced storage with Semantic Web data retrieved from files, network locations, databases, and/or constructed directly.

3. Combinations: The framework combines your referenced Semantic Web data from multiple places to create additions, unions, differences, and intersections as well as test for equality between the referenced locations.

4. Reasoning: The framework allows internal and external reasoning of the Semantic Web to produce additional information based on inference. The additional information could add new statements and also indicate issues with existing statements.

5. Interrogation: The framework investigates the Semantic Web data through searching, navigation, and queries. Searching uses simple matching. Navigation follows the path created by the various property relationships, and queries employ a formal query language.

6. Export: The framework provides methods to export the Semantic Web data in various standard formats.

7. Deallocation /close: The framework clears out the referenced storage and frees any allocated computing resources.

We have demonstrated a complete life cycle of using the Jena Semantic Web Framework classes and methods in dealing with Semantic Web data.

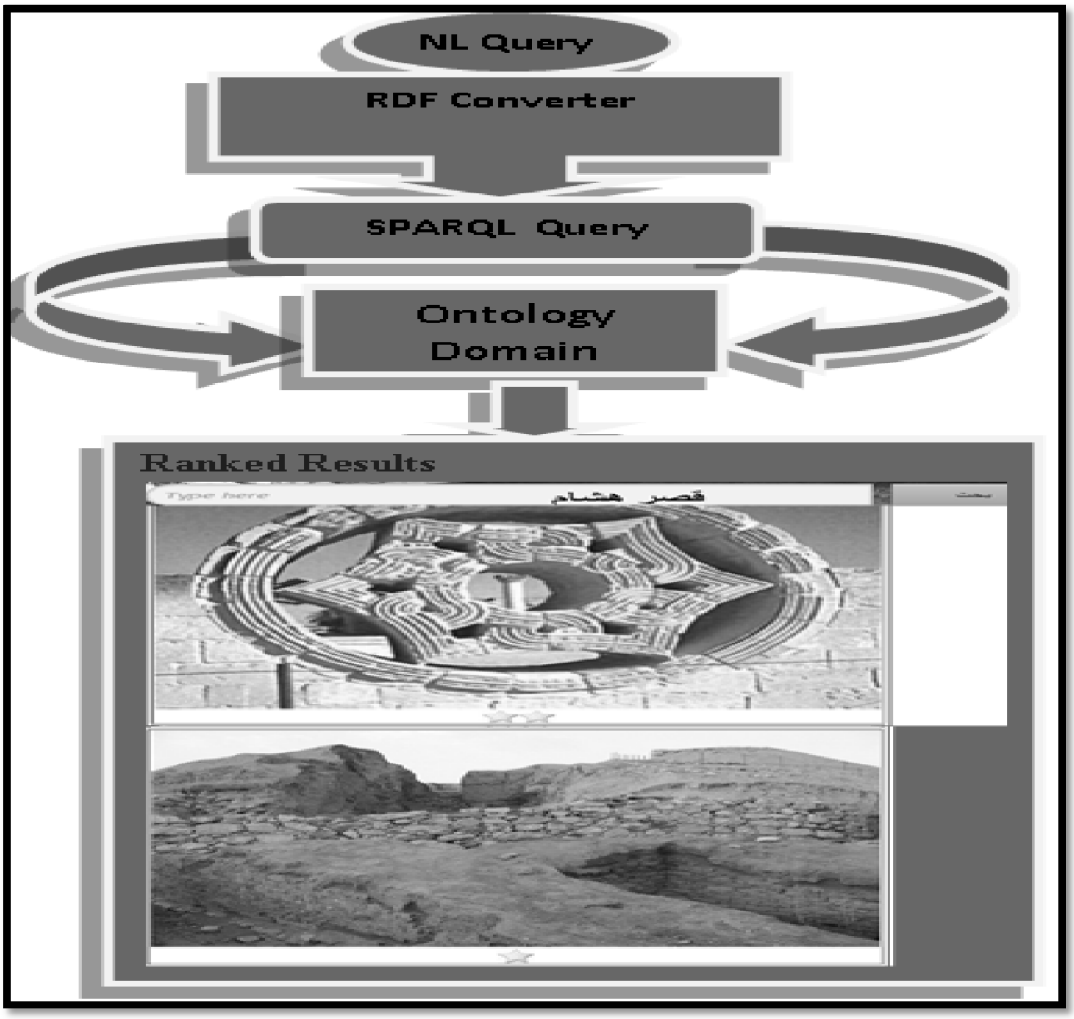


Figure 5.1: PalTourism prototype based on Ontology .

5.1.1 Jena Semantic Web Framework

We selected the Jena Semantic Web Framework because it strikes a useful balance between the various Semantic Web languages, offers excellent flexibility, and is open source. Jena is implemented in the Java programming language.

These Java-based abstractions translate the statements and constructs of the Semantic Web into useful programming artifacts such as Java classes, objects, methods, and attributes.

In addition to offering classes for typical Semantic Web constructs, Jena offers classes to convert ontologies to Java classes. Jena offers a Java class, schemagen, to generate a Java class description of a Semantic Web ontology or schema. This does not convert the instance data into Semantic Web data, only the ontology statements. It is limited to the ontology or schema constructs. The schemagen Jena class constructs a Java class for each of the Semantic Web classes, allowing your application programmatic access to its underlying components.

Typically, schemagen is called directly from a command window or via an ant script, a Java-based automated code-building tool. You need only provide a few options and the location of the ontology via a file or URL. As an example we use schemagen to generate a Java class from the domain ontology. Here is the command-line call.

```
java -classpath "./jena.jar:
./commons-logging-1.1.1.jar:./xercesImpl.jar:
./xml-apis.jar:./log4j-1.2.12.jar:
./iri.jar:./icu4j
3 4.jar"
```

```
jena.schemagen -i foaf.rdf --package
"net.semwebprogramming.chapter10.JenaExploration"
-o PalToursimowl.java ·ontology
```

The command-line call includes the necessary classpath .jar files as well as several key schemagen options:

- i—Provides the input location of the ontology. This can be either a local file or a remote URL. Here we have a local file, PalToursim.rdf.
- package—Provides the package name for the created Java class.
- o—Provides the name of the output file to contain the Java class.
- Ontology—The class uses OWL constructs as opposed to the default RDF constructs.

5.1.2 Ontology Creation:

Our Domain (PalToursim) Ontology Creation Method:

1. Determine the scope of the ontology. Domain Name: PalToursim (المواقع السياحية في فلسطين).
2. Consider reusing (parts of) existing ontology.(we didn't find any ready Arabic Ontology for tourism.)
3. Enumerate all the concepts we want to include(السياحة،المواقع الاثرية،المواقع الدينية،المواقع الطبيعية،).
4. Define the taxonomy of these concepts.(we define of the concepts and classes in the PalTourism prototype ontology. For each class, we define its super class and subclasses; give a human readable, non-normative description.)
5. Define properties of the concepts.

Table5.1.2: PalTousim Ontology Properties of the Concepts

• الاسم • المكان • الشخص الذي بناها • التاريخ • الصفات المميزة • الاحداث المرتبطة • الصورة • الاجزاء • الموقع	<ObjectProperty IRI="#له_حدث"/> </Declaration> <Declaration> <ObjectProperty IRI="#له_شخص_مسؤول"/> </Declaration> <Declaration> <ObjectProperty IRI="#له_صورة"/> </Declaration> <Declaration> <ObjectProperty IRI="#له_عنوان"/> </Declaration> <Declaration> <ObjectProperty IRI="#له_كيان"/> </Declaration> <Declaration> <ObjectProperty IRI="#له_موقع"/> </Declaration> <Declaration> <ObjectProperty IRI="#هو__جزء"/> </Declaration> <Declaration> <DataProperty IRI="#له_احداثيات"/> </Declaration> <Declaration> <DataProperty IRI="#له_صورة"/> </Declaration> <Declaration> <DataProperty IRI="#لها_اسم-عربي"/> </Declaration> <Declaration> <DataProperty IRI="#لها_اسم_انجليزي"/> </Declaration>

6. Define facets of the concepts such as cardinality, required values etc.(we define the relationship between all the classes, define the required entry type)
7. Define instances.(انواع الاماكن السياحية).

Table 5.1.2.1 : PalTousim Ontology Subclasses

<SubClassOf> <Class IRI="#تصنيف"/> <ObjectAllValuesFrom> <ObjectProperty IRI="#له_تصنيف"/> <ObjectUnionOf> <Class IRI="#ترفيهي"/> <Class IRI="#حضاري"/> <Class IRI="#ديني"/> <Class IRI="#طبيعي"/> </ObjectUnionOf> </ObjectAllValuesFrom>	○ مكان ديني • مسجد • مقام • كنيسة • دير ○ مكان أثري • قصر • متحف • برج • قاعة • بلدة قديمة • حمام • خان • قلعة ○ مكان طبيعي • منطقة طبيعية • ساحل • محمية طبيعية • منتجع طبيعي • حديقة

8. Mine Arabic concepts/glosses from dictionaries.("سياحة:ذهب وسار ورجع فيها من المشرق الى المغرب" "الاثري:القديم المأثور" "الديني:الدين:اسم لجميع ما يعبد به الله""الطبيعي:نسبة الى الطبيعة وهي طبائع الاشياء وما اختصت فيه من القوة")حسب شرح المعجم الوسيط-الطبعة الرابعة

5.1.3 Ontology Creation Tool

Our Domain ontology was created with Protégé (4.2 version), a graphical ontology-development tool: Supports a rich knowledge model, Open-source and freely available (http://protege.stanford.edu/index.shtml). Protégé is a tool which provides the following features to the user:

1. construction of a domain ontology
2. customization of data
3. entry of data

We also use special OWL-Plug-in, to create Arabic ontology called (Jambalaya 2.7.0) another special feature and extension of Protégé. This OWL-Plug-in provides the following features:

- Load and save OWL ontologies in Arabic Language
- Edit and visualize OWL classes and their properties

5.1.4. Customizing the Jena Framework:

The Jena Framework allows customized implementations to provide flexibility.

The model actually consists of a collection of objects that implement a Jena Graph interface. The Jena Graph interface provides a limited set of methods. Our application creates a Java class that implements the Jena Graph interface to customize graph behaviors such as altering the persistence implementation. Creating a custom model requires two steps:

1. The creation of the customized graph-based object

2. The creation of the model based on the customized Graph object.

Table 5.1.4: the basic structure of Jena Graph

```
Model customModel = null;

private void createCustomModel(){
CustomGraph myGraph = new CustomGraph();
customModel = ModelFactory.createModelForGraph(myGraph);
}
public class CustomGraph implements Graph {

public void close() {
}
public boolean contains(Triple arg0) {
return false;
}
public boolean contains(Node arg0, Node arg1, Node arg2) {

return false;
}
public void delete(Triple arg0) throws DeleteDeniedException {
}
public boolean dependsOn(Graph arg0) {
return false
}
public ExtendedIterator find(TripleMatch arg0) {
return null;
}
public ExtendedIterator find(Node arg0, Node arg1, Node arg2) {
return null;
}
public BulkUpdateHandler getBulkUpdateHandler() {
return new CustomBulkUpdateHandler();
}
public Capabilities getCapabilities() {
return null;

public BulkUpdateHandler getBulkUpdateHandler() {
return new CustomBulkUpdateHandler();
}
public Capabilities getCapabilities() {
return null;
}
public GraphEventManager getEventManager() {
return new CustomGraphEventManager();
}
public PrefixMapping getPrefixMapping() {
return new CustomPrefixMapping();
}
public Reifier getReifier() {
return new CustomReifier();
}
public GraphStatisticsHandler getStatisticsHandler() {
return new CustomGraphStatisticsHandler();
```

```
}
public TransactionHandler getTransactionHandler() {
return new CustomTransactionHandler();
}

public BulkUpdateHandler getBulkUpdateHandler() {
return new CustomBulkUpdateHandler();
}
public Capabilities getCapabilities() {
return null;
}
public GraphEventManager getEventManager() {
return new CustomGraphEventManager();
}
public PrefixMapping getPrefixMapping() {
return new CustomPrefixMapping();
}
public Reifier getReifier() {
return new CustomReifier();
}
public GraphStatisticsHandler getStatisticsHandler() {
return new CustomGraphStatisticsHandler();
}
public TransactionHandler getTransactionHandler() {
return new CustomTransactionHandler();
}
```

5.1.5 Serializing Semantic Web Data

Serialization offers the ability to transmit the model via various means and then reconstitute it on its reception. Models and Graph objects are not directly serializable. Nevertheless, you can take advantage of a model's capability to export a stream into a buffer that you can serialize. The following code illustrates the serialization:

Table 5.1.5: Serializing Semantic Web Data

```
public byte [] exportModel(){
ByteArrayOutputStream io = new ByteArrayOutputStream();
model.write(io);
return (io.toByteArray());
}
SerializableModel serialModel =
new SerialzableModel( model.exportModel);
public class SerializableModel implements Serializable {
private byte graphbuf[];
SerializableModel( byte graph[]){
graphbuf = graph;
}
```

```
public SemanticObject importModel(byte[] buf){
ByteArrayInputStream input = new ByteArrayInputStream(buf);
model.read(input,defaultNameSpace);
return this;
}
```

5.1.6. Exposing Jabber with a Custom Streaming RDF Writer

We Tried demonstrating the use of a streaming writer to generate RDF in the Turtle syntax. Jabber represents a useful tool to have when working with large volumes of data, and other more general-purpose approaches can't scale adequately to the task. We used to expose the Jabber Java client data source for (PalToursim prototype) as RDF. There are two main parts of this Process. First is a streaming Turtle writer class, TurtleWriter. This class provides methods for creating new individuals and then appending property values to them. The writer itself is relatively straightforward and doesn't have a lot of advanced features, but it demonstrates show a very simple concept can be used to expose a large amount of data in a scalable fashion. The second part of the code is the main application that uses the Jabber client to generate a Java representation of auser's contact list and his online status information and then iterate through that model, using the Turtle writer to generate RDF that reflects the properties and values of the data. Most of the code that performs this task is contained in a method of the (JabberToRdf)class called (retrieveToursim).

Table 5.1.6:RDF Converter

```
public InputStream retrieveFriends(String server, String user,
String pass)
{
InputStream toReturn = null;
XMPPConnection connection = null;
try
{
//create a connection
```

```
connection = new XMPPConnection(server);
//connect and log in
connection.connect();
//get the contact list
Roster roster = connection.getRoster();
roster.setSubscriptionMode(Roster.SubscriptionMode.accept
all);
Collection<RosterEntry> entries = roster.getEntries();
ByteArrayOutputStream baos = new ByteArrayOutputStream();
//create a turtle writer
TurtleWriter writer = TurtleWriter.createTurtleWriter(baos);
//add the prefixes for this document
writer.addPrefix("j", "http://www.jabber.org/ontology#");
writer.addPrefix("", "http://www.jabber.org/data#");
for(RosterEntry entry : entries)
{
//open the individual
writer.openIndividual("", entry.getUser(), "j", "Contact");
//write their name if they have one
if(null != entry.getName())
{
}
writer.addLiteral(
"rdfs", "label", entry.getName(), "xsd:string");
writer.addLiteral(
"j", "name", entry.getName(), "xsd:string");
//write their presence state
Presence p = roster.getPresence(entry.getUser());
String type = getType(p.getType());
String mode = getMode(p.getMode());
String status = p.getStatus();
if(null != type)
{
writer.addReference("j", "presenceType", "j", type);
}
if(null != mode)
{
writer.addReference("j", "presenceMode", "j", mode);
}
if(null != status)
{
```

```
writer.addLiteral("j", "status", status, "xsd:string");
}
writer.closeIndividual();
}
writer.close();
toReturn = new ByteArrayInputStream(baos.toByteArray());
}catch (XMPPException e){}
return toReturn;
```

The goal of this code is to demonstrate the use of the streaming writer, not necessarily to focus on Jabber. The first dozen of lines of the method establish the connection with the Jabber server and retrieve the user's contact list. The actual contact list is represented by the instance of Roster, and a snapshot of the contact list at a point in time can be generated by calling Roster.getEntries(). Jabber uses asynchronous message passing to establish the status of contacts on the contact list, so each time getEntries() is called, the list may be different. The code ignores the asynchronous model and generates an RDF representation of the client's contact list (roster) at a single point in time. The unabridged source code corresponding to the code above contains a thread sleep operation that gives the client five seconds to gather status information for the contacts in the roster. Once the roster snapshot is generated, the TurtleWriter is created, and then each entry in the roster is processed and written. The code that creates the TurtleWriter is as follows:

```
ByteArrayOutputStream baos = new ByteArrayOutputStream();
//create a turtle writer
TurtleWriter writer = TurtleWriter.createTurtleWriter(baos);
```

The rest of the loop writes the other properties of this individual, including name, presence type, presence mode, and status. Following is a very compressed listing of the rest of the properties that are output for the current entry in the roster:

```
writer.addLiteral("rdfs", "label", entry.getName(), "xsd:string");
writer.addLiteral("j", "name", entry.getName(), "xsd:string");
writer.addReference("j", "presenceType", "j", type);
writer.addReference("j", "presenceMode", "j", mode);
writer.addLiteral("j", "status", status, "xsd:string");.
```

This process is repeated for each entry in the roster, and the end result is an RDF graph serialized to Turtle that represents the current state of the user's contact list.

We reuse a custom RDF writer like this to expose any data source that can be worked with in Java.

5.1.7. Sparql for the Query

Table 5.1.7:SPARQL Query for the images in Ontology Domain

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbprop: <http://dbpedia.org/property/>
PREFIX foaf: <http://xmlns.com/PalToursim/0.1/>
SELECT ?picture
WHERE {
matches(?first, ?second, ?third, ?regex)
PalToursim:img ?picture
} ORDER BY ?matches(individual,property,equlavant)
```

5.2. The Second Prototype (Relational Database Based):

The search engine based on relational database life cycle follows these steps:

1. Parser Creation.
2. Relational Database Creation.
3. Matching Algorithm (between the Query and the relational database).
4. Interface Creation

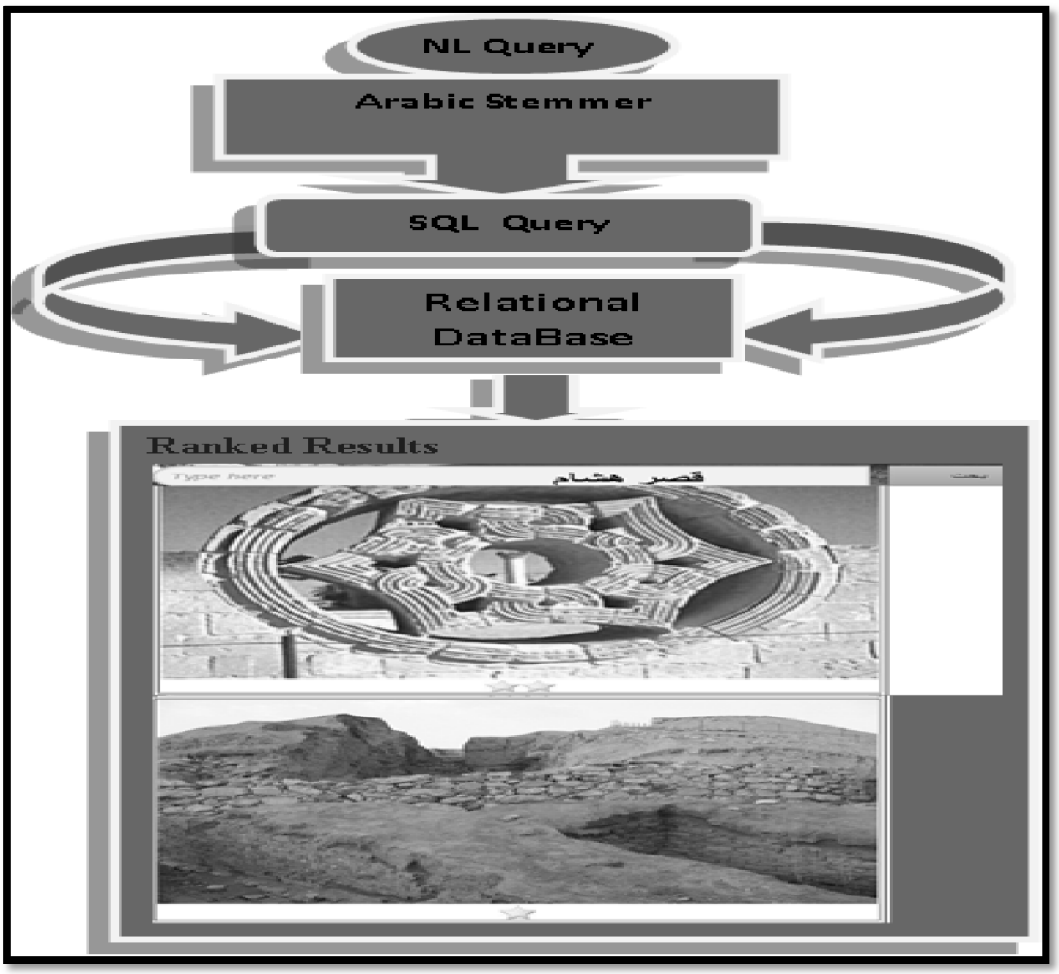


Figure 5.2: PalTourism prototype based on Relational Database

5.2.1 Arabic Parser Creation

Arabic Parser methodology included these steps:

1. Split the query to token according to the spaces between then words.

2. Remove all stop words, and all under 3 letters.
3. Then remove the prefix and the suffixes, if what is left of the word contains a valid stem-stop.
4. Once a stem is found, a pattern is then constructed by replacing the root letters from the remaining part of the word with the letters of the basic pattern.

*** For, lists of prefixes, suffixes, verb roots, solid word roots, patterns, foreign words, and function words were created using statistical studies (we use shereen khoja Lists as reference in verb roots).**

Table 5.2.1: Stemmer Algorithm

```
namespace PalTourismParser
{
    class PalParser
    {
        List<string> Roots= new List<string>();
        public void parse(string Statment)
        {string[] Tokens = Statment.Split(' ');
            string root = string.Empty;
            for (int index = 0; index < Tokens.Length; index++)
            { root = Tokens[index];
                if (root.Length < 3)
                    continue;
                if (root.Length > 3  && (root.StartsWith("وال")
||root.StartsWith("بال") || root.StartsWith("عال") ||
root.StartsWith("فال") || root.StartsWith("ال")))
                {  root= root.Replace("وال","");
                    if (root.Length > 3)
                    root= root.Replace("فال","");
                    if (root.Length > 3)
                    root= root.Replace("بال","");
                    if (root.Length > 3)
                    root = root.Replace("عال", "");
                    if (root.Length > 3)
                    root = root.Replace("ال", "");
                }if (root.Length > 3)
                {
                    root = root.Replace("ا", "");
                    if (root.Length > 3)
                    root = root.Replace("و", "");
                    if (root.Length > 3)
                    root = root.Replace("ي", "");
                    if (root.Length > 3)
                    root = root.Replace("أ", "");

                }
                if (root.Length > 3)
                {
                    if (root.EndsWith("ة"))
                        root = root.Remove(root.Length - 1, 1);
```

```
                }
                if (root.Length > 3)
                {
                    if (root.EndsWith("\"))
                        root = root.Remove(root.Length - 1, 1);
                }
                if (root.Length > 3)
                {
                    root = root.Remove(0, 1);
                }
                if (root.Length > 3)
                    root = root.Remove(root.Length - 1, 1);
                Roots.Add(root);

            }


            // Compose a string that consists of three lines.
            string lines = "First line.\r\nSecond line.\r\nThird
line.";

            // Write the string to a file.
            System.IO.StreamWriter file = new
System.IO.StreamWriter("D:\\Roots.txt");
            foreach (string root_item in Roots)
            {
                file.WriteLine(root_item);

            }
            file.Close();
```

5.2.2 Database Creation

We used Microsoft SQL Server 2012 to create our database structure which contains eight tables have clowns with same name ontology entities, to help us in the comparison process.

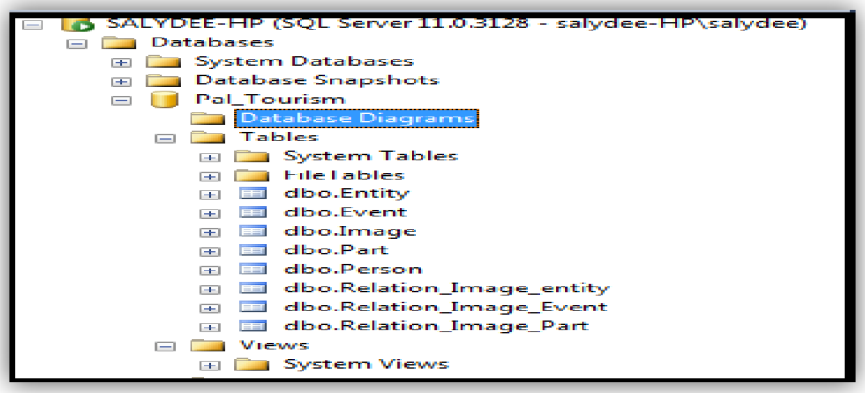


Figure 5.2.2: PalTourism Database Tables

	Column Name	Data Type	Allow Nulls
	[اسم الكيان]	nchar(10)	☑
	[تصنيف الكيان]	nchar(10)	☑
	[موقع الكيان]	nchar(10)	☑
	[اجزاء الكيان]	nchar(10)	☑
▶	[رقم الكيان]	int	☑

Figure 5.2.2.1: PalTourism Database Table Details

5.2.2.1 PalToursim Database Entry

we designed special page in (aspx) to enter the images and there metadata.

mage_ID	Title	E_title	Image			
	قبة الصخرة قبة الصخرة القدس	Dome of the Rock		Add entities	Add Parts	Add Events
	قبة الصخرة القدس	Dome of the Rock		Add entities	Add Parts	Add Events
	الجليل الاعلى	Upper Galilee		Add entities	Add Parts	Add Events
	الجامع الابيض الرمله	white Mousqe		Add entities	Add Parts	Add Events

Figure 5.2.2.1: Aspx page for entering Database

Table 5.2.2.1: Aspx page for entering Database Algorithm

```
namespace Pal_Tourism
{
    public partial class About : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void GridView1_SelectedIndexChanged(object sender,
EventArgs e)
        {

        }

        protected void GridView1_RowCommand(object sender,
GridViewCommandEventArgs e)
        {


            if (e.CommandName == "Insert")
            {
                Response.Redirect("Addentitiestoimage.aspx?Image_ID="
+
GridView1.Rows[int.Parse(e.CommandArgument.ToString())].Cells[0].Text)
;
            }
            else if (e.CommandName == "Insert_Part")
            {
                Response.Redirect("Add_Parts_To_Image.aspx?Image_ID="
+
GridView1.Rows[int.Parse(e.CommandArgument.ToString())].Cells[0].Text)
;

            }
```

Our dataset contained (300 pictures), dimension (460*250) with jpg extension, all of the images are about tourism places in Palestine.

5.2.3 .Matching Algorithms

5.2.3.1 String Matching Algorithm:

- We used Brute-Force Algorithm [67]. The brute-force pattern matching algorithm compares the pattern ***P*** with the text ***T*** for each possible shift of ***P*** relative to ***T***, until either:

- A match is found.
- All placements of the pattern have been tried

- Example of worst case:
  - ***T*** = ***الاموي***
  - ***P*** =***الاموي***

- **Algorithm** ***BruteForceMatch***(***T, P***)
- **Input** text ***T*** of size ***n*** and pattern , ***P*** of size ***m***
- **Output** starting index of a substring of ***T*** equal to ***P*** or -1, if no such substring exists.

Table 5.4.1: Brute-Force Algorithm

```
for  i ← 0 to n - m
  { test shift i of the pattern }
  j ← 0
  while j < m ∧ T[i + j] = P[j]
          j ← j + 1
  if  j = m
          return  i {match at i}
  else
          break while loop {mismatch}
return  -1 {no match anywhere}
```

5.2.3.2 Lexical Matching Algorithm

We used Jaccard similarity [66] coefficient to find the lexical matching between the query and the relational database for rating the results.

$$Jaccard(Set1, Set2) = | Set1 \cap Set2 | / | Set1 \cup Set2 |$$

String 1:
“ أين يقع باب العمود؟”
Set1 = {باب، العمود، يقع}
String 2:

“من ابواب القدس المشهورة باب العمود”
Set2 = { باب ،العمود، ابواب، القدس ،المشهورة}

2/6 = 0.33

Table 5.4.1: Jaccard Similarity Algorithm

```
For i := 0 to High (InputMatrix.Cells [0]) do
      Begin
        // directly convert to binary variables
        FirstVal  := Abs (InputMatrix.Cells [RunnerX, i]) >
aZeroThresh;
        SecondVal := Abs (InputMatrix.Cells [RunnerY, i]) >
aZeroThresh;

        If FirstVal And SecondVal THen
        Begin
          J11 := J11 + 1;
        end
        Else
        Begin
          If FirstVal Then J10 := J10 + 1;
          If SecondVal Then J01 := J01 + 1;
        end;
      end;

      Denominator := J01 + J10 + J11;
      If Denominator > 0 THen Quotient := J11 / Denominator
                         Else
                         Begin
                           Quotient := NaN;
                           dist_JaccardSimilarity := False;
                         end;
```

Chapter Six

Evaluation

The most two frequent and basic measures for information retrieval effectiveness are precision and recall. These are defined for the simple case where an IR system returns a set of documents for a query.

The measures of precision and recall concentrate the evaluation on the return of true positives, asking what percentage of the relevant documents has been found, and how many false positives have also been returned [71].

The advantage of having the two numbers for precision and recall is that one is more important than the other in many circumstances. Typical web surfers would like every result on the first page to be relevant (high precision), but have not the slightest interest in knowing let alone looking at every document that is relevant. In contrast, various professional searchers such as paralegals and intelligence analysts are very concerned with trying to get as high recall as possible, and will tolerate fairly low precision results in order to get it[46]. Individuals searching their hard disks are also often interested in high recall searches. Nevertheless, the two quantities clearly trade off against on another you can always get a recall of 1 (but very low precision) by retrieving all items for all queries! Recall is a non-decreasing function of the number of items retrieved. On the other hand, in a good system, precision usually decreases as the number of items retrieved is increased. In general we want to get some amount of recall while tolerating only a certain percentage of false positives [71].

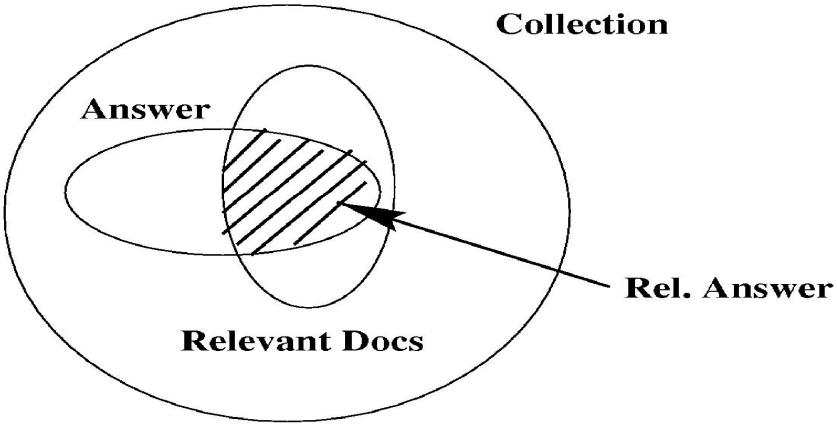


Figure 6.1: chart of Relevant Docs[71].

$$recall = \frac{Number\ of\ relevant\ Images\ retrieved}{Total\ number\ of\ relevant\ Images}$$

$$precision = \frac{Number\ of\ relevant\ Images\ retrieved}{Total\ number\ of\ Images\ retrieved}$$

- Precision
  - The ability to retrieve top-ranked items that are mostly relevant.
  - The fraction of the retrieved items that are relevant [72].
- Recall
  - The ability of the search to find *all* of the relevant items in the corpus.
  - The fraction of the relevant items that are retrieved [72].

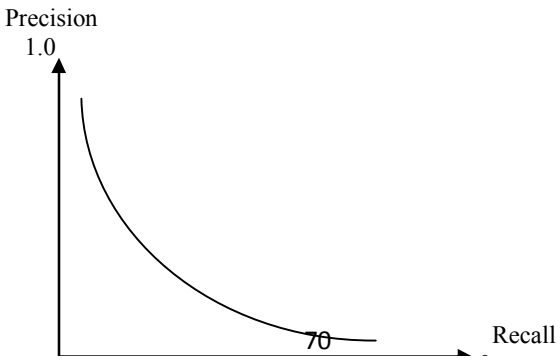


Figure 6.2: chart of Precision & Recall [71].

- Precision change w.r.t. Recall (not a fixed point)
- Systems cannot compare at one Precision/Recall point
- Average precision (on 11 points of recall: 0.0, 0.1, …, 1.0)

6.2. Evaluation Strategy

We used the (Gold Standard) strategy.

- Five experts' human marks(expert in tourism).
- Every human makes 50 queries, each time different query for different images.
- For a given query, produce the ranked list of retrievals.
- Adjust a threshold on this ranked list produces different sets of retrieved documents, and therefore different recall/precision measures.
- Mark each document in the ranked list that is relevant according to the query.
- Compute a recall/precision pair for each position in the ranked list that contains a relevant document.

Relevance of a document to an information need is treated as an absolute, objective decision. But judgments of relevance are subjective, varying across people, human assessors are also imperfect measuring instruments, susceptible to failures of understanding and attention. We also have to assume that users' information needs do not change as they start looking at retrieval results. Any results based on one collection are heavily skewed by the choice of collection, queries, and relevance judgment set.

6.2.1. Query Examples (Ontology Based Search Engine).

Example NO.1:

Query in Natural Language (كنيسة المهد في بيت لحم)

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

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

PREFIX foaf: <http://xmlns.com/PalToursim/0.1/>

SELECT ?picture

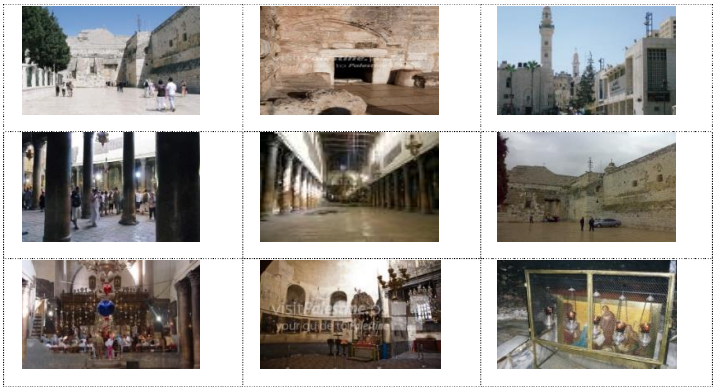
WHERE {

matches(?كنيسة ,?مهد ,? بيت ,?لحم)

PalToursim:img ?picture

} ORDER BY ?matches(individual,property,equlavant) **)**

Result:



Example NO.2:

Query in Natural Language (الحريق في الحرم الابراهيمي)

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

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

PREFIX foaf: <http://xmlns.com/PalToursim/0.1/>

SELECT ?picture

WHERE {

matches(?حريق ,?حرم ,?ابراهيمي)

PalToursim:img ?picture

} ORDER BY ?matches(individual,property,equlavant) **)**

Result:



6.3. Experimental Results for First Prototype (based on Ontology):

Table 6.3: Experimental Results for First Prototype based on Ontology

Query NO	Test No.1		Test No.2		Test No.3		Test No.4		Test No.5	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
1	1.00	0.02	1.00	0.03	1.00	0.03	1.00	0.03	1.00	0.03
2	1.00	0.05	1.00	0.05	0.50	0.03	1.00	0.05	1.00	0.05
3	1.00	0.07	1.00	0.08	0.67	0.06	1.00	0.08	0.67	0.05
4	1.00	0.10	1.00	0.10	0.75	0.09	1.00	0.10	0.75	0.08
5	1.00	0.12	1.00	0.13	0.80	0.13	1.00	0.13	0.60	0.08
6	1.00	0.14	1.00	0.15	0.83	0.16	1.00	0.15	0.67	0.10
7	0.86	0.14	1.00	0.18	0.71	0.16	0.86	0.15	0.71	0.13
8	0.75	0.14	1.00	0.21	0.75	0.19	0.88	0.18	0.75	0.15
9	0.89	0.17	0.89	0.23	0.89	0.22	0.89	0.18	0.89	0.18
10	0.80	0.19	1.00	0.26	0.80	0.25	0.80	0.20	0.80	0.21
11	0.82	0.21	0.91	0.26	0.73	0.25	0.73	0.20	0.73	0.21
12	0.75	0.21	0.92	0.28	0.67	0.25	0.75	0.23	0.75	0.23
13	0.69	0.21	0.85	0.28	0.69	0.28	0.77	0.25	0.69	0.23
14	0.71	0.24	0.79	0.28	0.71	0.31	0.79	0.28	0.71	0.26
15	0.67	0.24	0.80	0.31	0.67	0.31	0.80	0.30	0.67	0.26
16	0.69	0.26	0.75	0.31	0.69	0.34	0.81	0.33	0.69	0.28
17	0.65	0.26	0.76	0.33	0.65	0.34	0.82	0.35	0.71	0.31
18	0.89	0.29	0.89	0.33	0.89	0.38	0.89	0.38	0.89	0.31
19	0.63	0.29	0.74	0.36	0.68	0.41	0.84	0.40	0.68	0.33
20	0.65	0.31	0.70	0.36	0.65	0.41	0.85	0.43	0.70	0.36
21	0.67	0.33	0.71	0.38	0.67	0.44	0.86	0.45	0.71	0.38
22	0.68	0.36	0.73	0.41	0.68	0.47	0.86	0.48	0.73	0.41
23	0.70	0.38	0.70	0.41	0.70	0.50	0.87	0.50	0.74	0.44
24	0.71	0.40	0.71	0.44	0.71	0.53	0.83	0.50	0.75	0.46
25	0.72	0.43	0.72	0.46	0.72	0.56	0.80	0.50	0.76	0.49
26	0.73	0.45	0.73	0.49	0.73	0.59	0.81	0.53	0.73	0.49
27	0.89	0.48	0.89	0.51	0.89	0.59	0.89	0.55	0.89	0.51
28	0.75	0.50	0.75	0.54	0.71	0.63	0.79	0.55	0.71	0.51

29	0.76	0.52	0.72	0.54	0.69	0.63	0.79	0.58	0.72	0.54
30	0.77	0.55	0.73	0.56	0.70	0.66	0.80	0.60	0.73	0.56
31	0.77	0.57	0.74	0.59	0.68	0.66	0.81	0.63	0.74	0.59
32	0.78	0.60	0.75	0.62	0.66	0.66	0.81	0.65	0.75	0.62
33	0.79	0.62	0.76	0.64	0.67	0.69	0.82	0.68	0.76	0.64
34	0.79	0.64	0.74	0.64	0.68	0.72	0.82	0.70	0.76	0.67
35	0.80	0.67	0.74	0.67	0.69	0.75	0.83	0.73	0.77	0.69
36	0.89	0.69	0.89	0.69	0.89	0.78	0.89	0.75	0.89	0.72
37	0.81	0.71	0.76	0.72	0.68	0.78	0.84	0.78	0.78	0.74
38	0.82	0.74	0.76	0.74	0.66	0.78	0.82	0.78	0.79	0.77
39	0.82	0.76	0.74	0.74	0.64	0.78	0.79	0.78	0.77	0.77
40	0.83	0.79	0.75	0.77	0.65	0.81	0.80	0.80	0.78	0.79
41	0.83	0.81	0.73	0.77	0.63	0.81	0.78	0.80	0.78	0.82
42	0.83	0.83	0.74	0.79	0.64	0.84	0.79	0.83	0.79	0.85
43	0.84	0.86	0.74	0.82	0.63	0.84	0.77	0.83	0.79	0.87
44	0.84	0.88	0.75	0.85	0.61	0.84	0.77	0.85	0.80	0.90
45	0.89	0.90	0.89	0.87	0.89	0.88	0.89	0.88	0.89	0.90
46	0.85	0.93	0.76	0.90	0.61	0.88	0.78	0.90	0.78	0.92
47	0.85	0.95	0.77	0.92	0.62	0.91	0.79	0.93	0.79	0.95
48	0.85	0.98	0.77	0.95	0.63	0.94	0.79	0.95	0.79	0.97
49	0.84	0.98	0.78	0.97	0.63	0.97	0.80	0.98	0.78	0.97
50	0.84	1.00	0.78	1.00	0.64	1.00	0.80	1.00	0.78	1.00

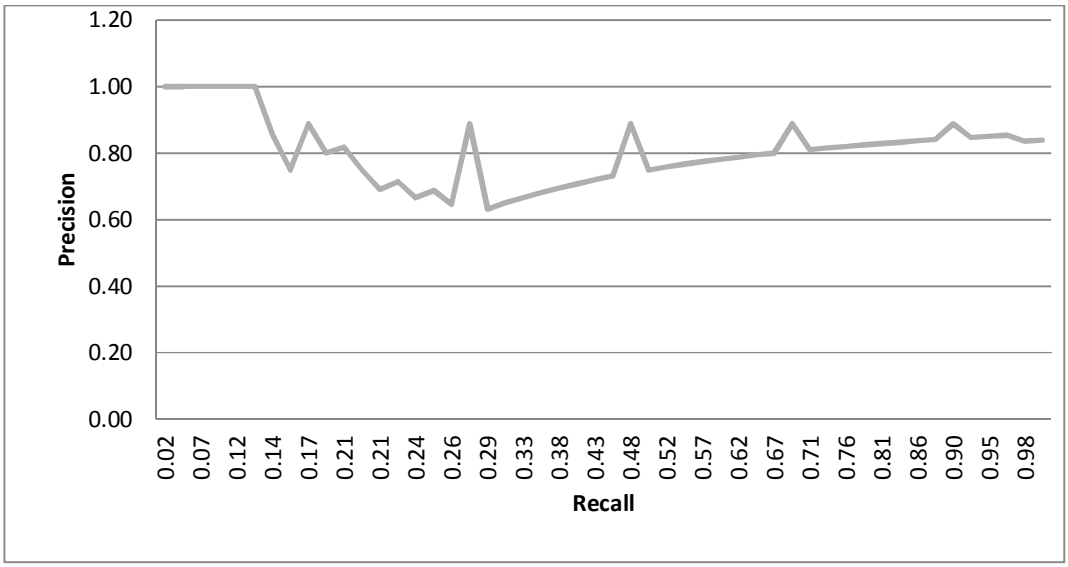


Figure 6.3.1: Chart (Test No.1) Averaged 50-point precision/recall

This curve graph represents 50 queries for a PalToursim search engine based on ontology for human mark (No.1). The Mean Average Precision for this system is (0.81); here we can see high value for precision which indicate the high performance of the system.

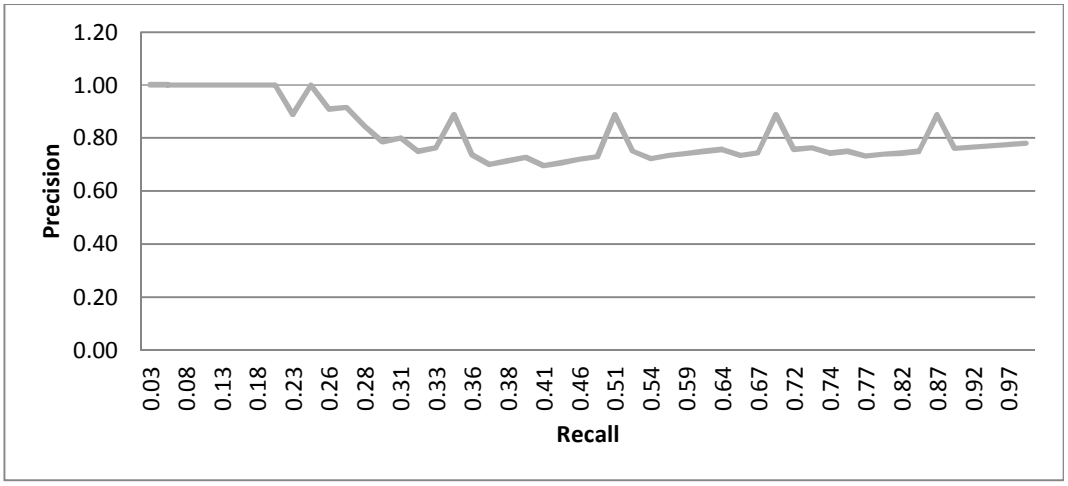


Figure 6.3.2: Chart (Test No.2) Averaged 50-point precision/recall

This curve graph represents 50 queries for a PalToursim search engine based on ontology for human mark (No.2). The Mean Average Precision for this system is (0.81) For a specific query, as we move down the result set, Precision of an initial segment goes up when we encounter a relevant document.

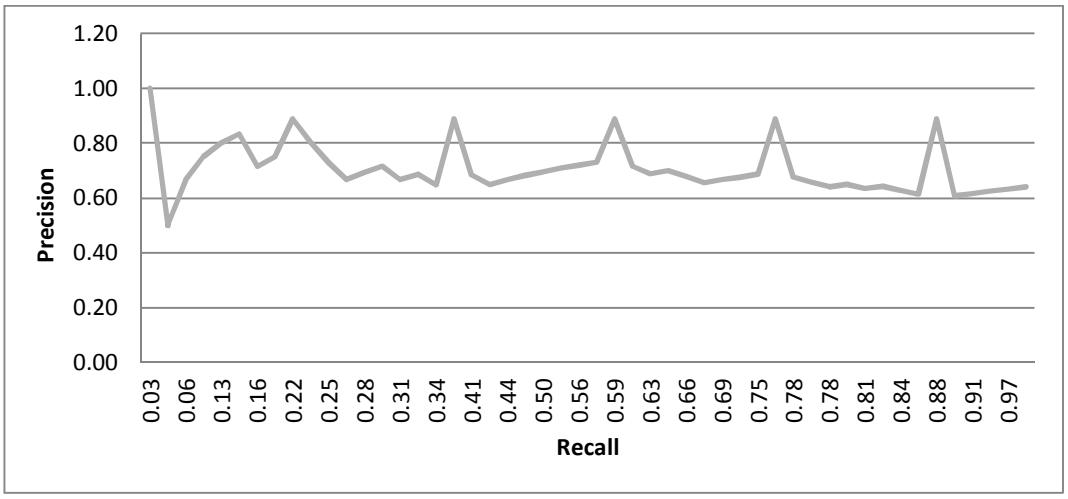


Figure 6.3.3: Chart (Test No.3)Averaged 50-point precision/recall

This curve graph represents 50 queries for a PalToursim search engine based on ontology for human mark (No.3). The Mean Average Precision for this system is (0.78) Higher precision early on is good for web search engines

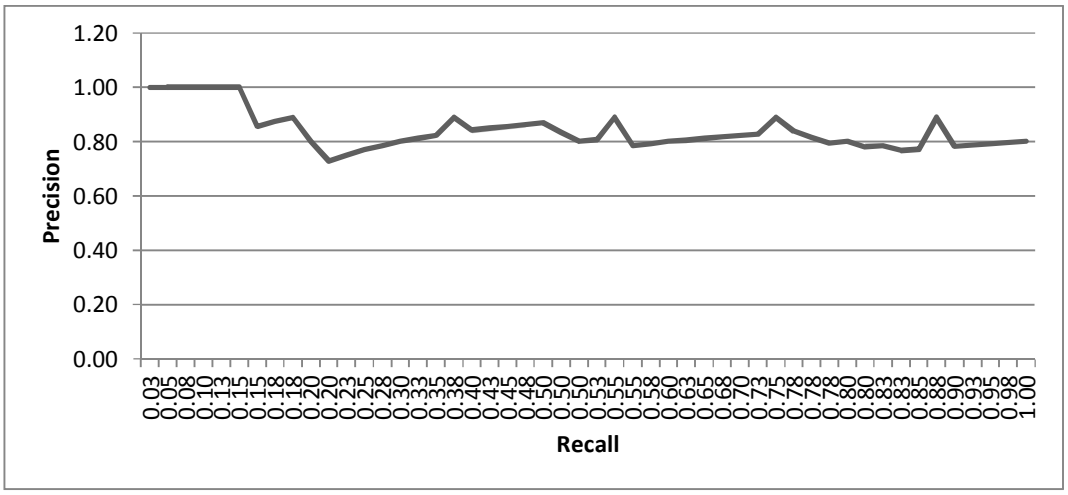


Figure 6.3.4: Chart (Test No.4) Averaged 50-point precision/recall

This curve graph represents 50 queries for a PalToursim search engine based on ontology for human mark (No.4). The Mean Average Precision for this system is (0.84)

Which is high precision value, and we can see recall of an initial segment goes up when we encounter a relevant document, and remains unchanged when we encounter an irrelevant document.

Figure 6.3.5: Chart (Test No.5) Averaged 50-point precision/recall

This curve graph represents 50 queries for a PalToursim search engine based on ontology for human mark (No.5). The Mean Average Precision for this system is (0.77). In a good system, precision decreases as either the number of docs retrieved or recall increase

6.3.1 Query Examples (Relational Database Based Search Engine).

Example NO.1:

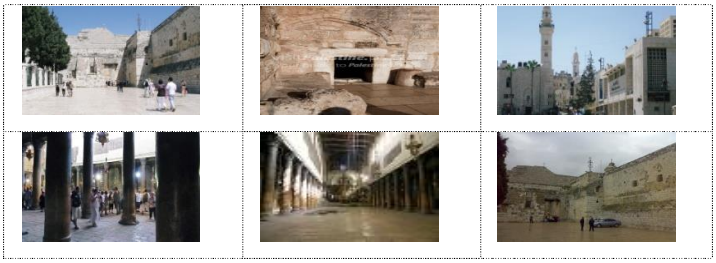
Query in Natural Language (كنيسة المهد في بيت لحم)

SQL Search

```
USE [Pal_Tourism]
GO

SELECT [Image_ID]
      ,[Title]
      ,[E_title]
      ,[Image]
  FROM [dbo].[Image]
  where title=(´ كنس%) and (´ مهد´) and("بيت%) and (´ لحم´)
GO
```

Result:



Example NO.2:

Query in Natural Language (الحريق في الحرم الابراهيمي)

SQL Search

```
 USE [Pal_Tourism]
GO

SELECT [Image_ID]
     ,[Title]
     ,[E_title]
     ,[Image]
  FROM [dbo].[Image]
  where title=(“ حرق”) and (“ حرم“) and(“ابراهيم”)
GO
```

Result:

6.4 Experimental Results for Relational Database:

Table 6.4: Experimental Results For Relational Database

Query NO	Test No.1		Test No.2		Test No.3		Test No.4		Test No.5	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
1	1.00	0.1	1.00	0.04	1.00	0.04	1.00	0.04	1.00	0.04
2	0.50	0.1	0.50	0.04	0.50	0.04	0.50	0.04	0.50	0.04
3	0.33	0.1	0.33	0.04	0.33	0.04	0.67	0.07	0.33	0.04
4	0.50	0.2	0.25	0.04	0.25	0.04	0.50	0.07	0.50	0.08
5	0.40	0.2	0.40	0.09	0.40	0.08	0.40	0.07	0.40	0.08
6	0.33	0.2	0.33	0.09	0.33	0.08	0.50	0.11	0.50	0.12
7	0.43	0.3	0.29	0.09	0.43	0.13	0.43	0.11	0.43	0.12
8	0.38	0.3	0.25	0.09	0.50	0.17	0.50	0.15	0.38	0.12
9	0.89	0.3	0.89	0.13	0.89	0.17	0.89	0.15	0.89	0.16
10	0.30	0.3	0.30	0.13	0.50	0.21	0.50	0.19	0.40	0.16
11	0.36	0.4	0.27	0.13	0.45	0.21	0.45	0.19	0.36	0.16
12	0.33	0.4	0.33	0.17	0.50	0.25	0.50	0.22	0.42	0.20
13	0.31	0.4	0.31	0.17	0.46	0.25	0.46	0.22	0.38	0.20
14	0.29	0.4	0.36	0.22	0.43	0.25	0.43	0.22	0.43	0.24
15	0.27	0.4	0.33	0.22	0.47	0.29	0.47	0.26	0.40	0.24
16	0.25	0.4	0.38	0.26	0.44	0.29	0.44	0.26	0.38	0.24
17	0.29	0.5	0.35	0.26	0.47	0.33	0.47	0.30	0.41	0.28
18	0.89	0.5	0.89	0.30	0.89	0.33	0.89	0.30	0.89	0.28
19	0.32	0.6	0.42	0.35	0.42	0.33	0.42	0.30	0.37	0.28
20	0.30	0.6	0.45	0.39	0.45	0.38	0.45	0.33	0.40	0.32
21	0.29	0.6	0.48	0.43	0.43	0.38	0.43	0.33	0.38	0.32
22	0.27	0.6	0.50	0.48	0.45	0.42	0.41	0.33	0.41	0.36
23	0.26	0.6	0.48	0.48	0.43	0.42	0.43	0.37	0.39	0.36
24	0.25	0.6	0.50	0.52	0.42	0.42	0.42	0.37	0.38	0.36

25	0.24	0.6	0.48	0.52	0.40	0.42	0.44	0.41	0.40	0.40
26	0.23	0.6	0.50	0.57	0.42	0.46	0.42	0.41	0.38	0.40
27	0.89	0.7	0.89	0.61	0.89	0.46	0.89	0.41	0.89	0.40
28	0.25	0.7	0.54	0.65	0.43	0.50	0.43	0.44	0.39	0.44
29	0.24	0.7	0.52	0.65	0.41	0.50	0.41	0.44	0.41	0.48
30	0.23	0.7	0.53	0.70	0.40	0.50	0.43	0.48	0.40	0.48
31	0.23	0.7	0.55	0.74	0.39	0.50	0.42	0.48	0.42	0.52
32	0.22	0.7	0.56	0.78	0.41	0.54	0.44	0.52	0.41	0.52
33	0.21	0.7	0.55	0.78	0.39	0.54	0.45	0.56	0.42	0.56
34	0.21	0.7	0.56	0.83	0.38	0.54	0.47	0.59	0.41	0.56
35	0.20	0.7	0.54	0.83	0.37	0.54	0.49	0.63	0.43	0.60
36	0.89	0.8	0.89	0.87	0.89	0.58	0.89	0.63	0.89	0.60
37	0.22	0.8	0.54	0.87	0.38	0.58	0.49	0.67	0.43	0.64
38	0.24	0.9	0.55	0.91	0.37	0.58	0.47	0.67	0.42	0.64
39	0.23	0.9	0.54	0.91	0.36	0.58	0.49	0.70	0.44	0.68
40	0.23	0.9	0.53	0.91	0.38	0.63	0.48	0.70	0.43	0.68
41	0.22	0.9	0.51	0.91	0.37	0.63	0.49	0.74	0.44	0.72
42	0.24	1	0.52	0.96	0.38	0.67	0.48	0.74	0.43	0.72
43	0.23	1	0.51	0.96	0.40	0.71	0.49	0.78	0.44	0.76
44	0.23	1	0.50	0.96	0.41	0.75	0.50	0.81	0.45	0.80
45	0.89	1	0.89	0.96	0.89	0.79	0.89	0.85	0.89	0.84
46	0.22	1	0.48	0.96	0.43	0.83	0.52	0.89	0.48	0.88
47	0.21	1	0.49	1.00	0.45	0.88	0.53	0.93	0.47	0.88
48	0.21	1	0.48	1.00	0.46	0.92	0.54	0.96	0.48	0.92
49	0.20	1	0.47	1.00	0.47	0.96	0.53	0.96	0.49	0.96
50	0.20	1	0.46	1.00	0.48	1.00	0.54	1.00	0.50	1.00

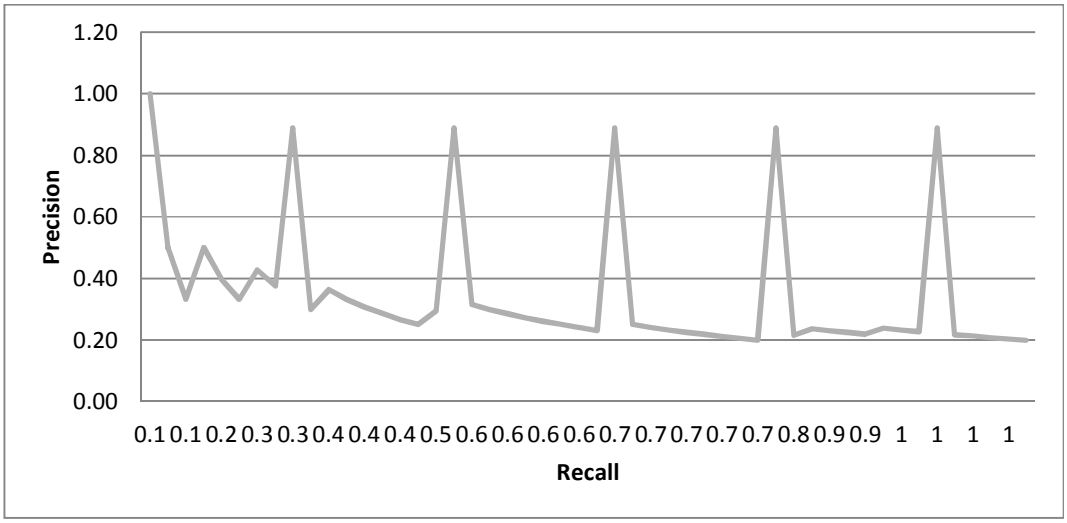


Figure 6.4.1: Chart (Test No.1)Averaged 50-point precision/recall

This curve graph represents 50 queries for a PalToursim system for human mark (No.1). The Mean Average Precision for this system is (0.39); here measures of precision and recall concentrate the evaluation on the return of true positives, representing percentages of the relevant documents have been found and how many false positives have also been returned. The advantage of having the two numbers for precision and recall is that one is more important than the other in many circumstances.

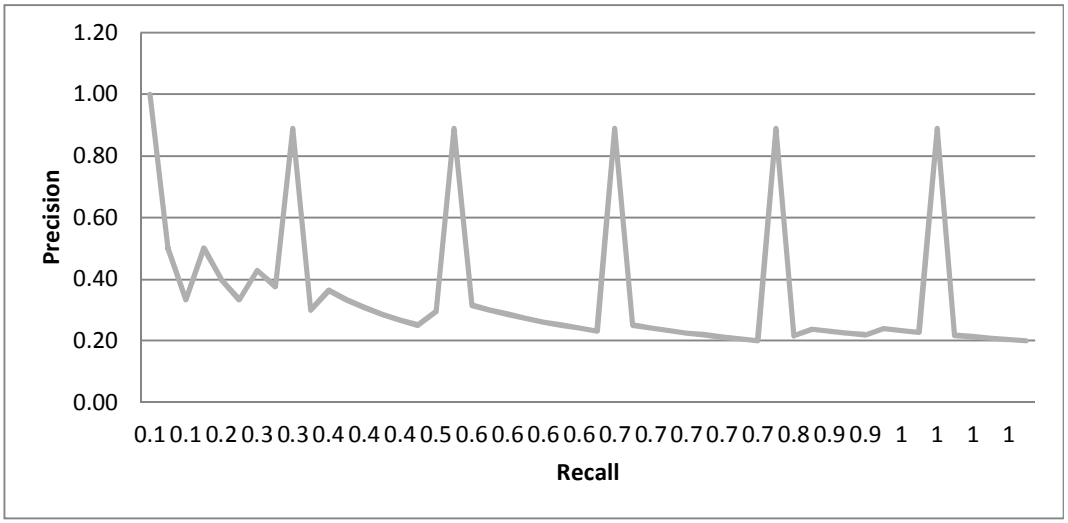


Figure 6.4.2: Chart (Test No.2)Averaged 50-point precision/recall graph

This curve graph represents 50 queries for a (PalToursim system) for human mark (No.2). The Mean Average Precision for this system is 0.50; Here we can always get a recall of 1 (but very low precision) by retrieving, all documents for all queries! Recall is a non-decreasing function of the number of documents retrieved.

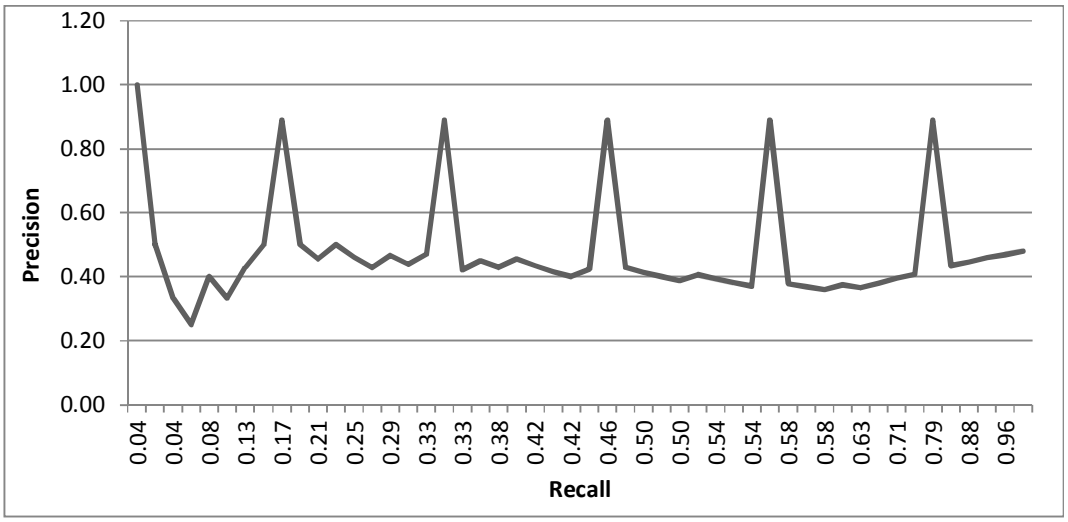


Figure 6.4.3: Chart (Test No.3) Averaged 50-point precision/recall

This curve graph represents 50 queries for a (PalToursim) search engine based on relational database for human mark (No.3). The Mean Average Precision for this system is 0.48; here you can see that precision usually decreases as the number of documents retrieved is increased. In general we want to get some amount of recall while tolerating only a certain percentage of false positives.

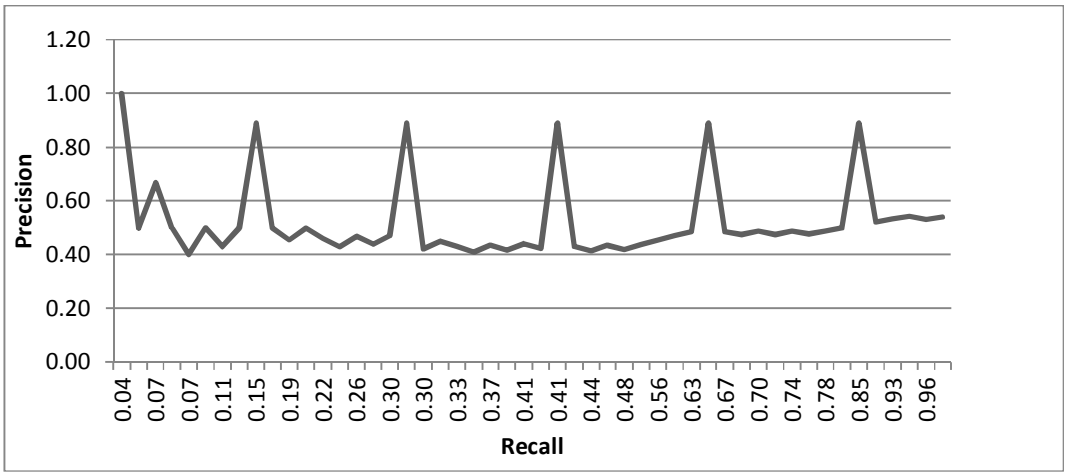


Figure 6.4.4: Chart (Test No.4) Averaged 50-point precision/recall

This curve graph represents 50 queries for a (PalToursim) search engine based on relational database for human mark (No.4). The Mean Average Precision for this system is(0.52).Precision-recall curves have a distinctive saw-tooth shape: if the (k + 1) image retrieved is no relevant then recall is the same as for the top k images, but precision has dropped. If it is relevant, then both precision and recall increase, and the curve jags up and to the right.

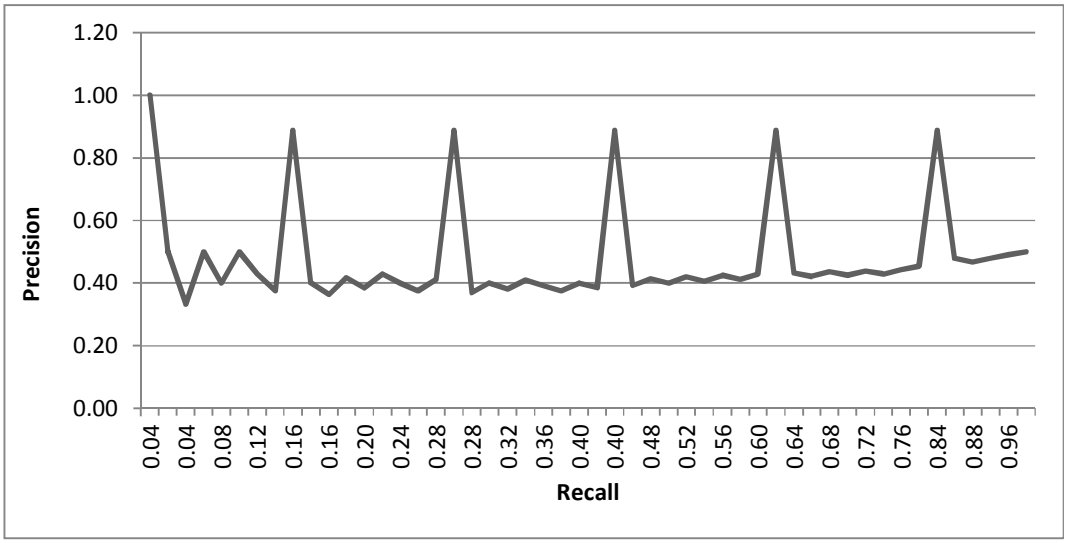


Figure 6.4.5: Chart (Test No.5) Averaged 50-point precision/recall

This curve graph represents 50 queries for a (PalToursim) search engine based on relational database for human mark (No.5). The Mean Average Precision for this system is (0.48). which is medium value according to high values we get from the other system which is base on Ontology.

6.5 . Comparison between the two results of the two systems:

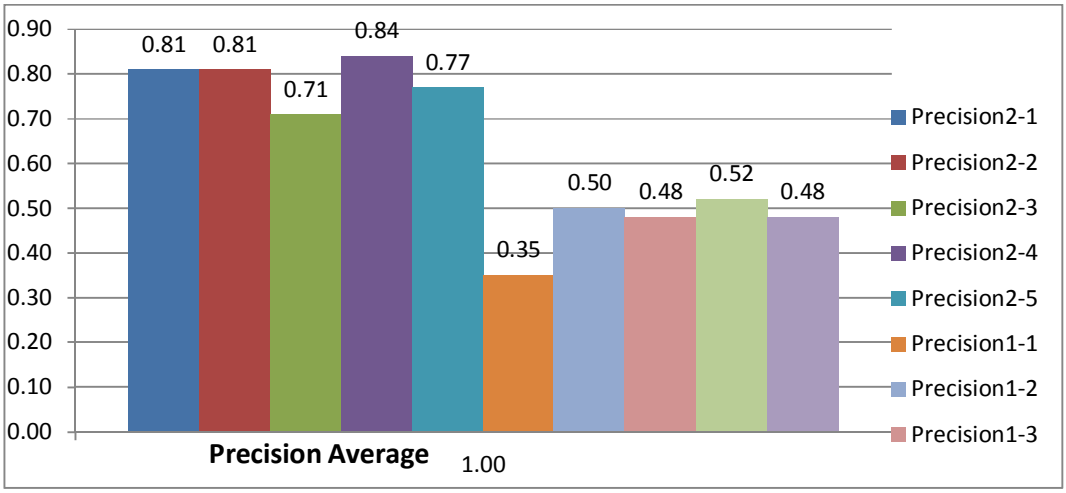


Figure 6.5.1: Chart Precision Average for the all tests for the both systems.

Here we can see how much the precision average for the system based on Ontology higher than the results for the system based on relational database, which indicate that the performance of the image retrieval system became higher when it based on Ontology ,because of the relational concepts on the ontology that make the retrieval more accuracy and more efficient**.**

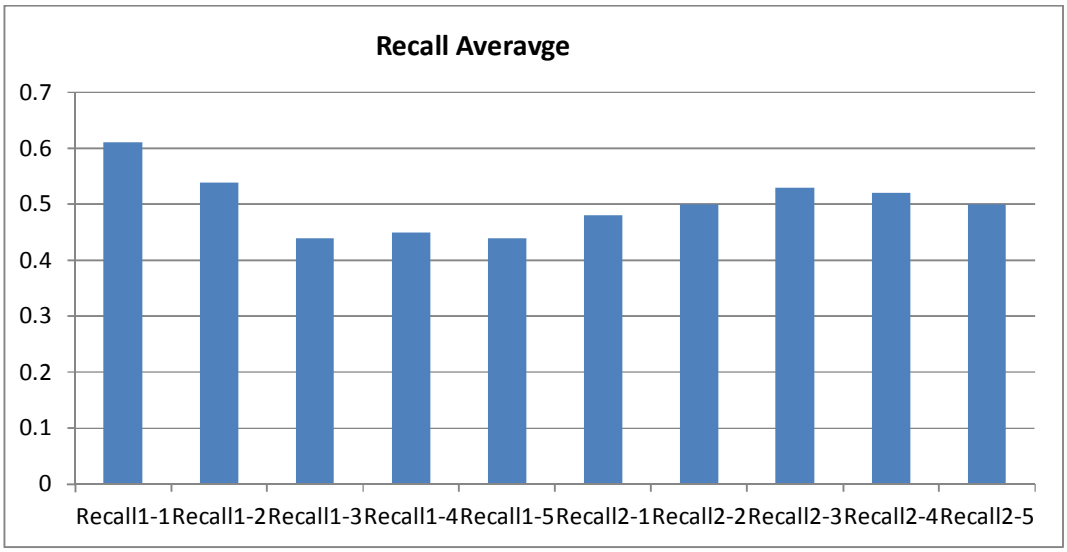


Figure 5.5.5: Chart Recall Average for the all tests for the both systems

Here we can see that Recall average for the all tests for the both systems are nearby because Recall fuction depends on retrieval all of the relevant items in the corpus. Returns most relevant documents but includes lot of junk. Recall is a non-decreasing function of the number of docs retrieved.

6.6. Conclusions & Future Work:

According to all our tests for our booth systems, the first one depends on the Arabic Domain Ontology we have built and the second one depends on the relational database we have also built with the same ontology classes and concepts.

We found that the system which depended on Ontology is more accurate, more efficient and more satisfying for the users. It has also higher average precision than the second one which told us of the efficiency of using Ontology of such systems for image retrieval.

Also we must mention that the formal evaluation measures are at some distance from our ultimate interest in measures of human utility: how satisfied is each user with the results the system gives for each information need that they pose. The standard way to measure human satisfaction is by various kinds of user studies. These might include quantitative measures, both objective, such as time to complete a task, as well as subjective, such as a score for the satisfaction with the search engine, and qualitative measures, such as user comments on the search interface.

A human concept is not a device that reliably reports a gold standard judgment of relevance of a document concerning a query. Rather, humans and their relevant judgments are quite idiosyncratic and variable. But this is not a problem to be solved:

in the final analysis, the success of our two systems depend on how good they are in satisfying the needs of these idiosyncratic humans, how much images are needed at a time.

Answering this question of whether IR evaluation results are valid despite the variation of individual assessors' judgments, people have experimented with evaluations taking one or the other of the two judges' opinions as the gold standard. The choice can make a considerable absolute difference to reported scores, but has in general been found to have little impact on the relative effectiveness ranking of either different systems or variants of a single system which are being compared for effectiveness.

The relevance of one document is treated as independent of the relevance of other documents in the collection. Relevance of a document to an information need is treated as an absolute, objective decision. But judgments of relevance are subjective, varying across people.

But we also took in our consideration during the evaluation of our system some critical points like:

• How fast does it retrieve, that is, how many images per second does it match?

Our both systems were fast in retrieving the required images.

• How expressive is its query language? How fast is the stemmer?

According to the confidence of the judges; both systems where expressive in Arabic language, but in system that depends on Ontology which is higher a little bit.

Our future work will be including more similar algorithms in matching queries to make the system more accurate, increasing the size of the dataset to make the system more efficient.

Also we can extend our scope of ontology to include GPS to describe the spatial aspect of the entities of the tourism locations**,** to make it available to integrate with e-maps or graphical directions, (like Google maps) .

References:

1. NikolaosSimo, Thanos Athanasiadi, Stefanos Kollias, Giorgos Stamou, and Andreas Stafylopatis, "Semantic Adaptation of Neural Network Classifiers in Image Segmentation", Springer-Verlag Berlin Heidelberg 2008.
2. "Web Services for Remote Portlets Specification", Approved as an OASIS Standard August 2003.
3. Lakshmi Palaniappan, N. Sambasiva Rao, G. V. Uma, "Development of Dining Ontology Based On Image Retrieval", International Journal of Scientific Engineering and Technology 2013.
4. Circuits and Systems for Video Technology200, IEEETrans.S.F. Chang et al., "special issue on MPEG-7 "
5. Astrova, Irina. "Toward the Semantic Web 'An Approach to Reverse Engineering of Relational Databases to Ontologies "Tallinn University of Technology, Ehitajate tee 2002.
6. Dongrong Xu, Chang Zhang ,Virginia Polytech "Analogical Reasoning for Information Retrieval" Eighth Americas Conference on Information Systems2002.
7. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, "Introduction to Information Retrieval", Online edition (c) 2009 Cambridge UP.
8. Hong Sun, Kristof Depraetere, Jos De Roo, Boris De Vloed,
9. Giovanni Mels, Dirk Colaert, "Semantic integration and analysis of clinical data" Advanced Clinical Applications Research Group, Agfa HealthCare, Gent, Belgium 2012.
10. Martin Doerr, "Ontologies for Cultural Heritage" , ICS-Forth, Greece2009.
11. Anne-Marie Tousch,phane Herbin, Jean-Yves Audibert, "Semantic hierarchies for image annotation" Elsevier Ltd2011.
12. Eero Hyv¨onen, Avril Styrman, and Samppa Saarela "Ontology-Based Image Retrieval", Helsinki Institute for Information Technology (HIIT) 2000.
13. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom,B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, "Query by image and video content: the QBIC system". MIT Press,Cambridge, MA, USA 1997.
14. Bach, J.R., Fuller, C., Gupta, A., Hampapur, A., Horowitz, B.,Humphrey, R., Jain, R., Shu, " Virage image search engine: Anopen framework for image management". Storage and Retrieval for Image and Video Databases (SPIE) 1996.
15. Frankel, C., Swain, M.J., Athitsos, V.: "Webseer: An image search engine for the world wide web". Technical report, The University of Chicago, Chicago, IL, USA 1996.
16. T. Gevers and A.W.M. Smeulders. PicToSeek: "Combining color and shape in-variant features for image retrieval". IEEE Transactions on Image Processing 2000.
17. J. R. Smith and S. F. Chang. Visually searching the web for the content. IEEE Multimedia Magazine, 1997.Alkhalifa, and Adam PEASE. "Introducing the Arabic WordNet Project"
18. Soo, V.W, Lee, C.Y, and Li. "Automated semantic annotation and retrieval based on sharable ontology and case-based learning techniques"

19. Saathoff, Timmermann, N., Staab, Petridis, and Anastasopoulos. " Linking ontologies with multimedia low-level features for automatic image annotation " Demos and Posters of the 3rd European Semantic Web Conference (ESWC)2006.
20. Hare, Sinclair, P.A.S, Lewis, and Martinez. "Mastering the Gap: From Information Extraction to Semantic Representation".
21. Srikanth, Varner, Bowden, and Moldovan. "Exploiting ontologies for automatic image annotation".
22. Athanasiadis, T., Tzouvaras, V., Petridis, K., Precioso, F., Avrithis,Y., Kompatsiaris, Y" Using a multimedia ontology infrastructure for semantic annotation of multimedia content" Proceedings of International Workshop on Knowledge Markup and Semantic Annotation(SemAnnot), Springer 2005.
23. Hirst, Graeme. "Introduction to Arabic Natural Language Processing".
24. Miller, G.A.: Wordnet: A lexical database for english. Communications of ACM 38(11) 1995.
25. Karin C. Ryding, A Reference Grammar of Modern Standard Arabic, Georgetown University, Karin C. Ryding 2005.
26. surveyImad A. Al-Sughaiyer; Ibrahim A. Al-Kharashi,
27. Arabic morphological analysis techniques: A comprehensive Journal of the American Society for Information Science and Technology; Feb 1, 2004.
28. Mohamed Maamouri, Ann Bies, Tim Buckwalter, Wigdan Mekki ,The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpu, Linguistic Data Consortium, University of Pennsylvania.
29. Spence Green and Christopher D. Manning, Better Arabic Parsing: Baselines, Evaluations, and Analysis, Computer Science Department, Stanford University2009.
30. Attia, Mohammed. "Handling Arabic Morphological and Syntactic Ambiguity within the LFG Framework with a View to Machine Translation"
31. Jarrar, Mustafa. "Building A Formal Arabic Ontology. In proceedings of the Experts Meeting On Arabic Ontologies And Semantic Networks"2010.
32. Llorente Coto., Ainhoa. "The Use of Ontologies for Improving Image Retrieval and Annotation".
33. Mizoguchi, Riichiro, Petasis George, Karkaletsis Vangelis, Krithara Anastasia, and Paliouras Georgios, "Semi-automated ontology learning"
34. Arpinar, Budak, Cartic Ramakrishnan, and Amit Sheth. "Geospatial Ontology Development and Semantic Analytics "
35. Berger, Tobias, Alissa Kaplunova, Atila Kaya, and Ralf Moller . "Towards a Scalable and Efficient Middleware for Instance Retrieval Inference Services"
36. Jarrar, Mustafa . "Towards The Notion Of Gloss, And The Adoption Of Linguistic Resources In Formal Ontology Engineering."
37. Klein. , Michel. "Change Management for Distributed Ontologies"
38. Malai, eronique, Laura Hollink , and Luit Gazendam. "The Interaction Between Automatic Annotation and Query Expansion: a retrieval experiment on a large cultural heritage archive".
39. Kesorn, Kraisak. "Semantic Restructuring of Natural Language Image Captions to Enhance Image Retrieval "
40. Iqbal, Qasim, and J. K. Aggarwal. "Using Structure in Content-based Image Retrieval".
41. Mensink, Thomas, Jakob Verbeek, and Gabriela Csurka. " Weighted Transmedia Relevance Feedback for Image Retrieval and Auto-annotation"

42. Kollias, Stefanos, Thanos Athanasiadis, Nikolaos Simou, Giorgos Stamou, and Andreas Stafylopatis. “Semantic Adaptation of Neural Network Classiïfiers in Image Segmentation”.
43. Espinosa Peraldi, Sofia, Atila Kaya, and Ralf M oller. "Formalizing Multimedia Interpretation based on Abduction over Description Logic Ontologies”.
44. Jassal, Azhar , and David Bell. "Fly-By-OWL: A Framework for Ontology Driven E-commerce Websites”.
45. Moller , Ralf, Volker Haarsle, and Michael Wessel . "On the Scalability of Description Logic Instance Retrieval”.
46. Mylonas, Phivos, Yannis Kalantidis, Giorgos Tolias, Christos Zigkolis, and Symeon Papadopoulos. "Image Clustering Through Community Detection on Hybrid Image Similarity Graphs”.
47. Espinosa, Soffia, Peraldi , and Atila Kaya . "Ontology and Rule Design Patterns for Multimedia Interpretation”.
48. G. Correndo, H. Alani, and P. Smart. A community based approach for managing ontology alignments.In Ontology Matching workshop , 2008.
49. K. Eckert, C. Meilicke, and H. Stuckenschmidt. Improving ontology matching using meta-level learning.In Proceedings of the 6th European Semantic Web Conference on The Semantic Web (ESWC-2009).
50. Hyvæonen, Eero, Avril Styrman, and Samppa Saarela. "Ontology-Based Image Retrieval " .
51. oaei.inrialpes.fr, "Automated support for evaluating alignment and matching algorithms Version 1.0” (2005).
52. S. Lew , Michael, Nicu Sebe , and Chabane Djeraba Lifl. "Content-based Multimedia Information Retrieval:State of the Art and Challenges" .
53. Marsza, Marcin. "Semantic Hierarchies for Visual Object Recognition"
54. Petasis, Pavlina Fragkou, Aris Theodorakos, and Vangelis Karkaletsis . "Segmenting HTML pages using visual and semantic information”.
55. Marsza, PH. Mylonas, D. Vallet, P. Castell, M. FERNA, and Ndez. "Personalized information retrieval based on context and ontological knowledge”.
56. Castano, Silvana , and Alïo Ferrara . "Enhancing Ontology Concept Design by Knowledge Discovery" .
57. Spyrou, Evaggelos, Yannis Kalantidis, Giorgos Tolias, Phivos Mylonas , and Stefanos Kollias. "Intelligent Content Retrieval using a Visual Vocabulary andGeometric Constraints”.
58. Y.Mori, H.Takahashi, and R.Oka. "Semantically Routing Queries in Peer-based Systems”.
59. P.Duygulu, K.Barnard, J.F.G. de Freitas, and D. A.Forsyth. "Object recognitionas machine translation: Learninga lexiconfor a ïndixed image vocabulary”.
60. J. Jeon, V. Lavrenko, R. Manmatha, and D. A.Forsyth. “Automatic image annotation and retrieval using cross-media relevance models”.
61. V. Lavrenko, R. Manmatha, and J. Jeon. "Amodel for learning the semantics of pictures”.
62. D.Metzler, and R. Manmatha. "An inference network approach to image retrieval”.
63. F. Monay, and D. Gatica-Perez. "On image auto-annotation with latent space models”.

64. Nikolaos Simou, Georgios Papadopoulos, Rachid Benmokhtar, Krishna Chandramouli, and Vassilis Tzouvaras. "Integrating Image Segmentation and Classification for Fuzzy Knowledge-".
65. Nizar Habash, Reem Faraj, and Ryan Roth, Syntactic Annotation in the Columbia Arabic Treebank.
66. Ahmed Nwesris, Abdusalam F. "Effective Retrieval Techniques for Arabic Text".
67. Corey a. HarPer, "The dublin core Metadata"
68. Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn and Supachanun Wanapu, "Using of Jaccard Coefficient for Keywords Similarity", Proceedings of the International MultiConference of Engineers and Computer Scientists 2013.
69. Chen, Aitao. "Building an Arabic Stemmer for Information Retrieval".
70. Palmisano, Ignazio, Luigi Iannone, Domenico Redavid, and Giovanni Semeraro. "Ontology Alignment Through Instance Negotiation:"
71. Tousch, Anne-Marie, Ste ´ phane Herbin, and Jean-Yves Audibert. "Semantic hierarchies for image annotation "
72. Rettinger, Achim, Uta L ¨ osch, Volker Tresp, Claudia d'Amato, and Nicola Fanizzi. "Mining the Semantic Web".
73. Christodoulakis, Stavros, Michalis Foukarakis, and Lemonia Ragia. "Spatial Information Retrieval from Images using Ontologies and Semantic Maps "
74. J. Madhaven, P. Bernstein, and E. Rahm. Gweneric Schema Matching with Cupid. Proceedings of the 27th VLDB Conference. 2001
75. Z. Xu, S. Zhang, And Y. Dong. Mapping Between Relational Database Schema And Owl Ontology For Deep Annotation. International Conference On Web Intelligence. 2006.
76. Evaluation Book, 2009 Cambridge University Press. "Evaluation in information retrieval".
77. Smeaton, A.F., Over, P., Kraaij, W.: Evaluation campaigns and trecvid. In: Proceedings of the 8th ACM international workshop on Multimedia information retrieval (MIR), New York, NY, USA, ACM(2006).
78. Clough, P., M uller, H., Sanderson, M.: The clef 2004 cross-language image retrieval track. In: Fifth Workshop of the Cross-Language Evaluation Forum (CLEF), Heidelberg, Germany, Lecture Notes in Computer Science (LNCS), Springer (2005)