

إقرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

Automatic Ontology-Based Document Annotation for Arabic Information Retrieval

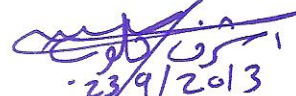
أقر بأن ما اشتملت عليه هذه الرسالة إنما هي نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وإن هذه الرسالة ككل، أو أي جزء منها لم يقدم من قبل لنيل درجة أو لقب علمي أو بحث لدى أية مؤسسة تعليمية أو بحثية أخرى.

DECLARATION

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification.

Student's name: **أشرف اسماعيل عبد القادر كلوب**

Signature:

التوقيع: 
23/9/2013

Date:

التاريخ: 2013/9/23م

بسم الله الرحمن الرحيم

Islamic University-Gaza
Deanery of Graduate Studies
Faculty of Information Technology



الجامعة الإسلامية- غزة
عمادة الدراسات العليا
كلية تكنولوجيا المعلومات

Automatic Ontology-Based Document Annotation for Arabic Information Retrieval

A Thesis Submitted as Partial Fulfillment of the Requirements for the Degree of
Master in Information Technology

Submitted By:
Ashraf I. Kaloub

Supervised by:
Dr. Rebhi S. Baraka

Zul-Qada 1434 H – September 2013



نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة عمادة الدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ أشرف اسماعيل عبدالقادر كلوب لنيل درجة الماجستير في كلية تكنولوجيا المعلومات برنامج تكنولوجيا المعلومات وموضوعها:

Automatic Ontology-Based Document Annotation for Arabic Information Retrieval

وبعد المناقشة التي تمت اليوم الاثنين 04 ذو القعدة 1434هـ، الموافق 2013/09/09م الساعة الثانية مساءً، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

مشرفاً ورئيساً

د. ربحي سليمان بركة

مناقشاً داخلياً

د. إياد محمد الأغا

مناقشاً خارجياً

د. يوسف نبيل أبو شعبان

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

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

والله ولي التوفيق،،،

مساعد نائب الرئيس للدراسات العليا

أ.د. فؤاد علي العاجز

Abstract

The rapid development in the semantic search technology gives motivation to build an efficient and scalable document annotation and retrieval techniques. Most existing methods and techniques in the field of document annotation and retrieval depend on English documents. Although the growing amount of Arabic content is being spread over the internet and other resources, there is little work carried out on Arabic semantic search and Arabic document annotation and retrieval.

In this research we propose an approach for enhancing the process of information retrieval for Arabic language that depends on the ontology in the process of document annotation. The results of the approach show significant improvement in the process of documents retrieval depending on the two common evaluation criteria precision and recall.

Keywords: Ontology, Information Retrieval, Document Annotation, Arabic Language.

عنوان البحث :

استعمال الأنتولوجيا للتمهيش الآلي للمستندات بقصد استرجاع المعلومات باللغة العربية

الملخص:

التطور السريع في مجال البحث الدلالي يعطي الدافعية لبناء تقنيات دلالية فعالة وقابلة للتطوير لاستخدامها في عملية استرجاع المعلومات، معظم الطرق والتقنيات الموجودة في مجال عمل التمهيش للمستندات واستخدامها في عملية استرجاعها تعتمد على المستندات باللغة الإنجليزية.

على الرغم من النمو المتزايد للمستندات العربية التي تنتشر عبر الإنترنت والمصادر الأخرى إلا إنه يوجد عمل قليل متعلق بالبحث باللغة العربية باستخدام الويب الدلالي.

هذا البحث يقدم طريقة لتحسين عملية استرجاع المستندات العربية بالاعتماد على الأنتولوجيا (Ontology)، حيث أن نتائج وتقييم الطريقة تعتمد على معيارين الدقة (Precision) والإسترجاع (Recall).

الكلمات الدالة: الأنتولوجيا، استرجاع المعلومات، تمهيش المستندات، اللغة العربية

Acknowledgements

Thanks and praise to Allah Almighty for guidance and help to complete this thesis. This thesis would not exist without the help, advice, inspiration, and encouragement of many people. I would like to thank my supervisor Dr. Rebhi Baraka for his time, patience, and understanding. I would also like to thank him for his advice during the period of study and his support on the general direction of this thesis and for the many questions he asked me to verify that I'm still on the right side.

Besides my advisor, I would also like to thank my colleagues and friends who encouraged me through my thesis work. My final words go to my family I want to thank my family, whose love and guidance is with me in whatever I pursue.

Ashraf I. kaloub

September 2013

Dedication

This thesis is dedicated to my dear parents, to my wife and children: Nour, Farrah and Ismail; without their love and support, this work would not have been possible.

Table of Contents

Abstract.....	I
المخلص:	II
List of Figures.....	VII
List of Tables.....	VIII
Chapter 1: Introduction.....	1
1.1 Statement of the Problem	2
1.2 Objectives	2
1.3 Importance of the Research	3
1.4 Scope and limitations of the research	3
1.5 Methodology.....	3
1.6 Thesis Structure	5
1.7 Summary.....	5
Chapter 2: Background.....	6
2.1 Semantic Web.....	6
2.1.1 Ontology	7
2.1.2 RDF and RDFS.....	11
2.1.3 OWL.....	12
2.2 Semantic Annotation	13
2.3 Semantic Annotation Tools	14
2.4 Information Retrieval	15
2.5 Semantic Search Engines.....	16
2.5.1 Semantic Search Engines Examples.....	18
2.6 Summary.....	18
Chapter 3: Related Work.....	19
3.1 Ontology-Based Annotations and Retrieval	19
3.2 Ontology-based Search and Retrieval	20
3.3 Summary.....	22
Chapter 4: The Proposed Model.....	23
4.1 Preparing the Corpus	23
4.2 Ontology Building	24
4.3 Documents Annotation	32
4.4 Processing Annotated Documents	34
4.5 Indexing and the Search Process	35
4.6 The Model Structure.....	36
4.7 Summary.....	37

Chapter 5: System Realization, Experimental Results and Evaluation	38
5.1 System Realization	38
5.1.1 Tools and Programs	38
5.2 System Interface	38
5.3 Experiments	40
5.4 System Evaluation	45
5.5 Discussion.....	47
5.6 Summary.....	48
Chapter 6: Conclusion and Future Work	49
References:	51
Appendix A: Part of OWL Source Code	55
Appendix B: Ontology Graph.....	62
Appendix C: Example of Jape Rules	63

List of Figures

Figure 2.1: A Layered Model of the Semantic Web Vision [26].....	6
Figure 2.2: A Small Ontology about Animal [35].....	7
Figure 2.3: Part of the OWL Code for Animal Ontology in Figure 2 [35].....	12
Figure 2.4: Semantic Annotation [20]	13
Figure 2.5: Information Retrieval Processes [52].....	16
Figure 2.6: Hakia [28]	17
Figure 2.7: Dogpile [28].....	17
Figure 4.1: Execution Steps.....	23
Figure 4.2: Part of the Ontology Concepts	27
Figure 4.3: Part of Ontology Concepts and Instances	28
Figure 4.4: Ontology Instances.....	29
Figure 4.5: The Synonyms Words for Instance "الجنابة"(Funeral)	30
Figure 4.6: Using Onto Root Gazetteer	33
Figure 4.7: The Annotation Process Result.....	33
Figure 4.8: Jape Rules Example	35
Figure 4.9: The Proposed Model	36
Figure 5.1: Gate Interface.....	39
Figure 5.2: Search Using Annotation Type "صلاة أهل الأعذار"(Prayer of Exempted People).....	40
Figure 5.3: Search Using Annotation Type "رواتب" (Roateb).....	41
Figure 5.4: Search Using Annotation Type "الأذان" (Aladan).....	43
Figure 5.5: Search Using the Word "الأذان" (Aladan) as Traditional Way.....	44
Figure 5.6: Recall and Precision for Every Annotation Type.....	46

List of Tables

Table 2.1: An Overview of Approaches to Ontology Evaluation [45].....	10
Table 4.1: Ontology Concepts	25
Table 4.2: Ontology Object Properties	29
Table 4.3: Ontology Evaluation.....	32
Table 5.1: Precision and Recall Results	46

Chapter 1: Introduction

Semantic web techniques have recently emerged as a new and highly promising context for knowledge and data engineering [1]. The semantic web is defined as "a group of technologies and approaches that allow machines to read, understand and retrieve the meaning of a specific "semantic" or information on the internet" [2]. The need for semantically enriched Information Retrieval (IR) and searching are among the most important issues of the semantic web. They try to overcome the limitations of the traditional IR model which suffers from misunderstanding the query and its context and on the keyword which cannot represent the semantic information of resources therefore obtaining a lower recall and precision.

Semantic search considers the meaning of words and phrases. Meaning is represented in machine readable format through an ontology which is formalized using Semantic Web languages such as Resource Description Framework (RDF) [3] and Web Ontology Language (OWL) [4]. By understanding the meaning of a query the results returned to the user will be more relevant. The IR and searching process can depend on document annotation, where semantic annotation involves tagging documents with concepts (e.g., ontology classes) so that content becomes meaningful [5]. Annotations can help users to easily organize their documents. Also, they can help in providing better search facilities, users can search for information not only using keyword-based search, but also using well-defined general concepts that describe the domain of their information need [6].

Most of the proposed methods or approaches in IR depend on English documents. Little work has been carried out on Arabic documents annotation and retrieval. Challenges in this regard include the lack of technology support and the lack of adequate support for semantic web tools [7].

Using ontology in the field of IR improves the retrieval accuracy and reduces irrelevant results. An Ontology is a formal explicit description of concepts in a domain of discourse classes (concepts). Properties of each concept to describe various features and attributes of the concept (slots), and restrictions on slots (facets) ontology together with a set of individual instances of classes constitutes a knowledge base [8].

In this research we develop an automatic ontology-based document annotation and retrieval method to support the process of IR for Arabic documents. We design a model that depends on ontology components which include classes, sub-classes, instances and relations to make

annotations. Also we enrich the ontology with a lot of synonyms and stemming word for ontology components to increase the process of documents annotation.

The current ontology-based document annotation systems is studied, ontology for a specific domain is designed and created, and the proposed ontology-based document annotation model and system is developed and evaluated.

1.1 Statement of the Problem

Although there is a large content of Arabic information found on the Web and other resources, little efforts have been exerted in the semantic-based Arabic document annotation and retrieval. The problem of this research is how to build an Arabic ontology-based document annotation and retrieval system that achieves the acceptable accuracy.

1.2 Objectives

▪ Main objective

The main objective of this research is to build an Automatic Arabic ontology-based document annotation and retrieval system which achieves the acceptable accuracy.

▪ Specific objectives

The specific objectives of this research are:

- To study the current ontology-based document annotation for information retrieval approaches to determine the requirements of an approach for Arabic.
- To build a domain-specific Arabic ontology to be used in the document annotation and retrieval process.
- To design the information annotation and retrieval system model.
- To implement the system prototype that uses the ontology for the process of document annotation.
- To conduct appropriate experiments with the annotated documents for document retrieval and searching.
- To evaluate the system for accuracy based on appropriate evaluation strategy.

1.3 Importance of the Research

- This research stems from the rapid increase of Arabic content on the Web and other resources and the need to extract important information from this content.
- The need for appropriate tools, techniques for Arabic documents.
- Information retrieval is very important to save time and effort when we search in a large repository or database that contain Arabic documents.
- Researchers in the field of semantic web can use the system as a basis for designing more specialized and more advanced systems for information retrieval in Arabic.
- The developed ontology can also be used as a basis for other applications and systems.
- The system can be used as a resource to depend on for inter organizational data integration between systems of different organizations.

1.4 Scope and limitations of the research

- The research will focus on file formats such as: doc and txt and will convert them into Extensible Markup Language (XML) type during the annotation and retrieval process.
- The annotation and retrieval process will depend on documents that exist in a corpus and not on the web.
- Only a prototype will be implemented to provide a proof of concept for the proposed approach. The prototype is tools specific and is not a standalone.
- Domain specific ontology is used for the process of annotation corpus of documents related to the issue of "فقه الصلاة" (Prayer jurisprudence).
- The used ontology does not cover the entire domain of "فقه الصلاة" (Prayer jurisprudence).
- The search process depends on the annotation types (ontology classes).

1.5 Methodology

Many researches and models have been proposed for developing ontology-based documents annotation and retrieval for the English language. In our work we address this problem by building ontology-based document annotation and retrieval model for Arabic language, through achieving the objectives stated in section 1.2 to achieve the objectives of the research the following methodology is followed:

- Studying and analyzing systems and applications that are related to document annotation and retrieval in different languages (Arabic and English).

- Preparing a corpus that contains the documents which will use, in the process of annotation and retrieval.
- Building Arabic domain ontology: we use a candidate tool (i.e. Protégé¹ [9]) to build the ontology. The development of ontology includes the following tasks:
 - Define concepts, i.e., classes based on studying and analyzing the domain.
 - Define instances, i.e., real elements in our domain.
 - Define relations among classes as a requirement to come up with the ontology.
 - Enrich ontology with Synonyms and Stemming for its components.
 - Ontology Evaluation
- Document annotation: we do this for a corpus using the designed ontology. This tests how well and appropriate the ontology is.
- Processing annotated documents is done using Jape rules² [42].
- Indexing and search process is done with the aid of Lucene³ [10] within Gate⁴ [11].
- Implementing a prototype of the proposed method by:
 - Specifying the requirements of the system.
 - Specifying the interaction between prototype components within Gate.
- Evaluate the approach and verify that it achieves the required accuracy. The method of verification is performed according to the two relevancy criteria namely precision and recall:
 - **Precision** is defined as the number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search.
 - **Recall** is defined as the number of relevant documents retrieved by a search divided by the total number of existing relevant documents (which should have been retrieved).

¹Protégé: is a free, open source ontology editor and knowledge-base framework.

²Jape rules: is a java annotation patterns engine provide finite state transduction over annotations based on regular expressions.

³Lucene: is a high-performance, full-featured text search engine library written entirely in Java.

⁴Gate: is open source software capable of solving almost any text processing problem.

1.6 Thesis Structure

The thesis consists of six chapters organized around the objectives of the research.

Chapter 1 introduces the research, problem, objectives, importance, scope and limitations, and methodology followed in the work.

Chapter 2 focuses on the background and theoretical concepts related to the domain of semantic web includes its technologies and techniques, semantic annotation tools, and semantic search engines.

Chapter 3 is devoted to related works and efforts in ontology building, information retrieval systems.

Chapter 4 presents the steps to execute the methodology and the architecture of the proposed model.

Chapter 5 is devoted to the evaluation of the proposed model and discussing the experiments and the findings.

Chapter 6 concludes the thesis and states future work.

1.7 Summary

In this chapter, we have introduced the thesis by giving an introduction about semantic web techniques and the terminologies related to it. We stated the research problem and questioning the possibility of using ontology in the process of Arabic documents annotation and retrieval by achieving the required accuracy. We stated the main objective of this research which is to automatic ontology based document annotation for Arabic information retrieval. Additionally we explained the importance of this research, which consider the system as a basis for designing more specialized and more advanced systems for information retrieval in Arabic. We also stated the scope and limitation of this research, the limitation is that the system model depends on documents that exist in a corpus and not in the web, and we build a system prototype not a complete system. We presented the methodology that will be followed in this research including preparing documents, building ontology, annotation documents, processing annotated documents, indexing and searching process and the evaluation of the model, implementing a prototype of the proposed method and evaluating the method. In the sixth section, we explained the thesis structure. Next chapter presents a background on semantic web, ontology building, semantic annotation tools, and semantic search engines.

Chapter 2: Background

This chapter presents the background and theoretical concepts of the Semantic Web and its technologies including ontology, RDF, RDFS, OWL, Semantic Annotation and its tools, Information Retrieval, Semantic Search engines needed in this research.

2.1 Semantic Web

The semantic web is defined as "a group of technologies and approaches that allow machines to read, understand and retrieve the meaning of a specific (semantic) or information on the internet [2]. The technological idea behind semantic web was derived from the ability to enable software and automated agents to access the Web effectively and intelligently [2]. Figure 2.1 depicts Tim Berners Lee's vision to the semantic web. It presented the basic framework of semantic web as a hierarchical structure whose function is improved layer by layer, at the bottom we find XML, a language that used to write structured web documents with a user-defined vocabulary. In the top of XML layer we find RDF layer which is a basic data model for writing simple statements about web objects. RDF Schema layer provides modeling primitives for organizing web objects into hierarchies and can be used as a primitive language for writing ontologies. The ontology layer is used to clearly represent objects and also relationships among them. A relationship may be direct or inverse. Additionally using ontologies helps machines to process the meaning and facilitate sharing of information. The logic layer is used to enhance the ontology language further and to allow the writing of application-specific declarative knowledge. The Proof layer involves the actual deductive process as well as the representation of proofs in Web languages and proof validation. The Trust layer will emerge through the use of digital signatures and other kinds of knowledge [47].

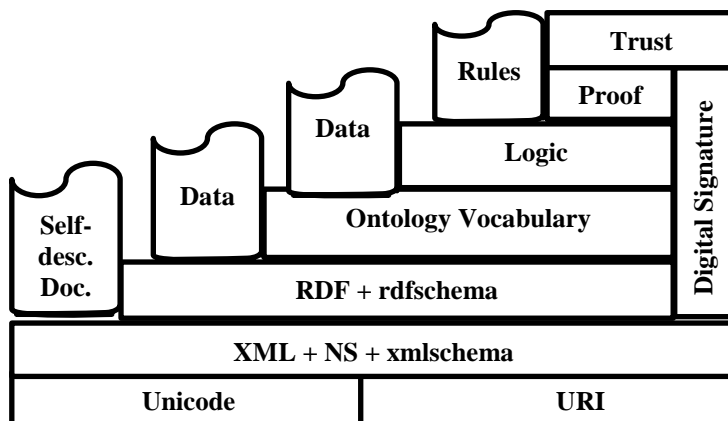


Figure 2.1: A Layered Model of the Semantic Web Vision [26].

In fact the Semantic Web does not differ from the World Wide Web but it is an extension of it. It is referred to as Web 3.0, the semantic web is a web of data, the semantic web is about

two things: it is about integration and combination of data from diverse sources and about how the relation of ontology to the semantic web [34].

2.1.1 Ontology

One of the best known definitions of ontology used in research on Artificial Intelligence and Knowledge Representation is "An ontology is an explicit specification of a conceptualization "[33]. Ontology is often represented as a set of concepts, relations, functions, and instances. Figure 2.2 is an example of a small ontology about animals. In the figure there is a class animal which have multiple object properties include eating, epidemic, habitation and protection. From the figure the object property eating make a triple between class animal and class food and the object property epidemic make triple between class animal and class disease. This can also be noticed for the two other object properties habitation and protection. Also the class animal have sub-class panda based on the relation is-a and the sub-class panda have multiple object properties include eating, epidemic, protection and habitation all of these object properties make triples between panda class and other classes for example object property eating make triple between panda and bamboo and object property protection make triple between panda and punishment.

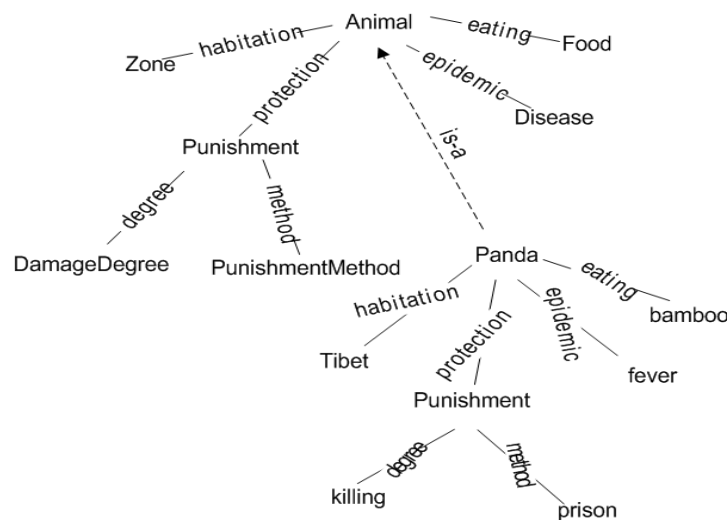


Figure 2.2: A Small Ontology about Animal [35]

Ontologies play an increasingly important role in knowledge management and is used as a standard knowledge representation for the Semantic Web. By ontology the users can connect with each other using a common understanding of a domain. Although ontology has been emerged as an important and natural means of representing real-world knowledge for the development of information systems, most ontology buildings are performed manually. This means ontology construction is a difficult and time-consuming task [36].

▪ **Ontology Building Methodologies**

Ontology building is not a simple task; it needs time, effort and expertise in the domain in which we want to build the ontology. There are a lot of methodologies for building an ontology; for example: SENSUS, KACTUS, OTK, CommonKADS, Tove, Methontology, Mikrokosmos, ONIONS and HYSSYS. All of these methodologies have their shortages: Some of these methodologies designed for integration of terms and ontology related to specified area cannot be used in other areas.. There are also some methods of the construction of ontologies using basic methods of ontology composition based on cases of experience. These cannot be regarded as structural construction methods. There are many methodologies that remain in a pre-development stage and do not include feasibility study. They are not helpful in solving problems occurring in real construction [48]. For manual construction of ontology we follow mainly Noy and McGuinness methodology [8]. It includes the following:

1. Determine the domain and scope of the ontology
2. Consider reusing existing ontologies
3. Enumerate important terms in the ontology
4. Define the classes and the class hierarchy
5. Define the properties of classes-slots
6. Define the facets of the slots
7. Create instances

Like any development process, above steps are not linear and will have to be iterated and backtracked to earlier steps at any point in the process [47].

Step 1. Determine the domain and scope of the ontology

This step defines the domain and the purpose of the ontology. Developing an ontology is not an aim or a goal in itself but we build the ontology for a particular purpose. This stage includes multiple and basic questions to be answered: what is the domain that the ontology will cover? For what we are going to use the ontology? For what types of questions should the ontology provide answers? Who will use and maintain the ontology? [47].

Step 2. Consider reusing existing ontologies

Find ontologies in the domain or the subject we work, so in most cases we don't need to begin from scratch in building our domain ontology. We can obtain our ontology from third-party that provide us this ontology [47], this will save a lot of time for us.

Step 3. Enumerate important terms in the ontology

This step is considered as the first step or the actual definition of the ontology where we make a list of an expected terms that will be used on the ontology building. It is important to get a comprehensive list of these terms without fear of overlap between concepts they represent or relations among the terms.

Step 4. Define the classes and the class hierarchy

After the identification of the relevant terms, these terms have to be organized in a hierarchical way, There are several possible approaches in developing a class hierarchy: A top-down development process starts with the definition of the most general concepts in the domain and subsequent specialization of the concepts. For example, we can start with creating classes for the general concepts. Then we specialize the class by creating some of its subclasses and so on. A bottom-up development process starts with the definition of the most specific classes then leaves of the hierarchy with subsequent grouping of these classes into more general concepts. For example, we start by defining classes then we create a common superclass for these classes. A combination development process of the top-down and bottom-up approaches [47] is also possible.

Step 5. Define the properties of class-slots

In this step, the classes that are created in the previous step does not provide enough information alone. So once we have selected the defined classes in the list of terms we created in Step 3, most of the remaining terms consider properties (slots) of these classes [8]. Where for each property in the list, we have to show which class it describes.

Step 6. Define the facets of the slots

In this step we add facets to the properties these facets include value type, allowed values, the number of the values (cardinality), and other features of the values the slot can take.

Step 7. Create instances

The last step is creating individuals (instances) of classes in the hierarchy, defining an instance of a class requires: (1) choosing a class, (2) creating an instance of that class, and (3) filling in the slot values.

▪ **Ontology Evaluation**

The built ontology should be evaluated using some evaluation criteria. These criteria can be divided into two types namely generic criteria and specific criteria. The generic criteria deals with factors like clarity, consistency and reusability. The specific criteria checks the generated ontology against the purpose and user requirements [49]. Most evaluation approaches fall into one of the following broad categories [45].

1. Comparing the ontology to a golden standard, may be the ontology itself.
2. Using the ontology in an application and evaluating the results
3. Comparing with a source of data (e.g. a collection of documents) about the domain to be covered by the ontology.
4. Evaluation is done by humans who try to assess how well the ontology meets a set of predefined criteria, standards and requirements.

Table 2.1: An Overview of Approaches to Ontology Evaluation [45].

Level	Golden standard	Application-based	Data-driven	Assessment by humans
Lexical, vocabulary, concept, data	x	x	x	x
Hierarchy, taxonomy	x	x	x	x
Other semantic relations	x	x	x	x
Context, application		x		x
Syntactic	x			x
Structure, architecture, design				x

The Golden standard evaluation: the Golden standard could be another ontology, based on a document-corpus or provided by domain experts. In Table 2.1 The lexical content of an ontology can also be evaluated using the concepts of precision and recall, in this case, the precision would be the percentage of the ontology lexical entries that also appear in the golden standard, relative to the total number of ontology words. Recall is the percentage of the golden standard lexical entries that also appear as concept identifiers in the ontology, relative to the total number of golden standard lexical entries. The same approach used to evaluate the lexical content of an ontology on other levels, e.g. instances, relations [45].

Application based evaluation: where the ontology will be used in some kind of application or task. The outputs of the application, or its performance on the given task, might be better or worse depending on the ontology used in it. So one might say that a good ontology is one which helps the application in question produce good results on the given task. Ontologies may therefore be evaluated simply by plugging them into an application and evaluating the results of the application [45].

Data-driven evaluation: An ontology may also be evaluated by comparing it to existing data (usually a collection of textual documents) about the problem domain to which the ontology refers [45].

Assessment by humans evaluation: this evaluation is done by humans who try to assess how well the ontology meets a set of predefined criteria, standards, requirements [45].

2.1.2 RDF and RDFS

Resource Description Framework (RDF) statement (or RDF triple) is of the form [subject property object.]. RDF-annotated resources (i.e., Subjects) are usually named by Uniform Resource Identifier references (URIs). RDF annotates Web resources in terms of names properties. The values of named properties (i.e., Objects) can be URIs of Web resources or literals, namely, representations of data values (such as integers and strings). A set of RDF statements is called an RDF graph. RDF Schema (RDFS) can be shown as a first try to support expressing simple ontologies with RDF syntax. In RDFS, predefined Web resources `rdfs: Class`, `rdfs: Resource`, and `rdf: Property` can be used to declare classes, resources, and properties respectively. From the first view, RDFS is a simple ontology language that supports only class and property hierarchies, as well as domain and range Restrictions for properties [31]. Basically, RDFS is built on top of RDF, and OWL is built on top of RDFS, any OWL ontology can be serialized using one of RDF formats also RDFS allows you to express the

relationships between things by standardizing on a flexible, triple-based format and then providing a vocabulary which can be used to say things. OWL is similar, but bigger, better. OWL lets you say much more about your data model, it shows you how to work efficiently with database queries and automatic reasoners, and it provides useful annotations for bringing your data models into the real world.

2.1.3 OWL

The Web Ontology Language (OWL) is as a standard (W3C recommendation) for expressing ontologies in the Semantic Web. The OWL language facilitates greater machine understandability of Web resources than that supported by RDFS by adding additional constructors for building class and property descriptions (vocabulary) and new axioms (constraints), along with a formal semantics [31]. Figure 2.3 shows a snippet of corresponding OWL code for ontology in Figure 2.2.

```
<owl:Class rdf:about="#Plant">
  <owl:disjointWith rdf:resource="#Animal"/>
</owl:Class>
<owl:Class rdf:ID="Animal">
  <owl:disjointWith>
    <owl:Class rdf:ID="Plant"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="Panda">
  <rdfs:subClassOf rdf:resource="#Animal"/>
</owl:Class>
...
<owl:Class rdf:ID="Food"/>
...
```

Figure 2.3: Part of the OWL Code for Animal Ontology in Figure 2 [35].

OWL ontologies are categorized into three species or sub-languages, namely OWL-LITE, OWL-DL, and OWL-FULL [31]. OWL Lite supports those needing a classification hierarchy and simple restrictions. OWL DL supports those who want the maximum expressiveness of the reasoning system. It includes all OWL language elements, but there are certain restrictions, such as the type of separation. It describes the logic of the existing technology is compatible with good reasoning computability. OWL Full supports maximum expressiveness, supports for the free use of RDF syntax, but does not have the computability of reasoning. There are a lot of differences between these three languages at the following. Capacity and reasoning ability in the expression, each sublanguage is the language of the previous

extension. Every legal OWL Lite ontology is a legal OWL DL ontology. Every legal OWL DL ontology is a legal OWL Full ontology. Every valid OWL Lite conclusion is a valid OWL DL. Every valid OWL DL conclusion is a valid OWL Full.

The developers can use the OWL ontology language that is suitable for their needs. Selecting OWL Lite or OWL DL depends on the extent to which users need to OWL DL provides most expressive elements [32].

2.2 Semantic Annotation

Semantic annotation refers to the process of tagging or annotating documents using ontology, so data becomes meaningful. An annotation can be carried out manually, but since this is very expensive in terms of user time annotating data can help in providing better search facilities, since it helps users to search for information not only based on the traditional keyword-based search, but also using well-defined general concepts that describe the domain of their information need [6]. Figure 2.4 depicts part of Kim ontology which shows the process of mapping between ontology classes with words in the text that achieve mapping for example, the classes Bulgaria, XYZ mapped with words Bulgaria, XYZ in the text.

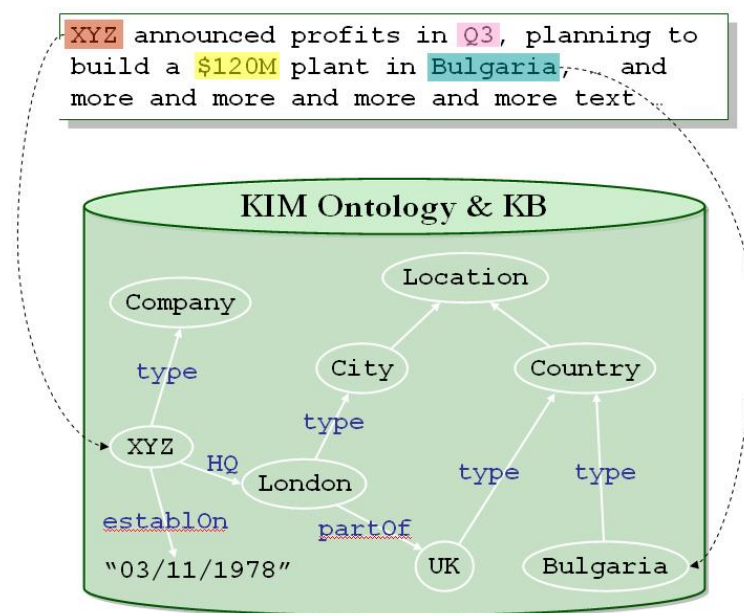


Figure 2.4: Semantic Annotation [20]

2.3 Semantic Annotation Tools

There are many tools available for semantic annotation of textual document like GATE [21], KIM [22], MnM [23], Magpie [24] etc. In sequence, we present a brief overview of all of these tools.

- **GATE**

GATE [21] is an architecture, a framework for Language Engineering (LE). Gate components of the three types: (1) Language Resources (LRs): represent entities such as lexicons, corpora or ontologies (2) Processing Resources (PRs): represent entities that are primarily algorithmic such as parsers (3) Visual Resources (VRs): represent visualization and editing components that participate in GUIs. GATE architecture is useful to develop a number of successful applications for various language processing tasks (such as Information Extraction Gate). Gate provided with a set of reusable processing resources for common NLP tasks these packaged together to form A Nearly-New Information Extraction system (ANNIE). ANNIE consists of the following main processing resources: tokenizer, sentence splitter, POS tagger, gazetteer, finite state transducer and orthomatcher these resources communicate via Gate annotation API. ANNIE relies on finite state algorithms and the JAPE language.

Also the GATE system provides many functionalities it provides the functionality to annotate textual documents both manually and automatically by running some processing resources over the corpus. Since manual annotation is a difficult and may cause error Gate tries to make it simple. To add a new annotation, select the text with a mouse (e.g., "Mr. Clever") and then click on the desired annotation type (e.g., Person), which is shown in the list of types on the right-hand-side of the document viewer. Gate can be used to develop applications and resources in multiple languages, based on its Unicode support.

- **KIM**

Knowledge and Information Management Platform (KIM) [22] is another ontology base semantic annotation system that uses KIM Ontology (KIMO). KIM also uses GATE, and Lucene information retrieval engine for many information extraction tasks. KIM has a feature by automatically adding new instances found in the text to Ontology. It also performs disambiguation step because many instances can be added to different places in ontology.

- **MnM**

MnM [23] an ontology annotation tool which provides both automated and semi-automated support for annotating web pages with semantic contents. MnM integrates a web browser with an ontology editor and provides open APIs to connect to ontology servers and for integrating information extraction tools. MnM can be seen as an new example of the next generation of ontology editors.

- **Magpie**

Magpie [24] is extended to the internet explorer enable ontology based semantic markup system that annotates the web pages. It uses ontology to annotate the web page either using a predefined lexicon in the ontology or using a Named Entity recognition technique. Some other system that is also used for semantic annotation are Onto-Mat [25] (work like MnM.).

2.4 Information Retrieval

Information retrieval (IR) is finding documents of an unstructured nature (usually text) that achieves an information need from within large collections (usually stored on computers) [28]. The traditional keyword-based IR technique performs keyword searching documents by matching the keywords that users determined in their queries. The systems maintain a word index to accomplish searching, a well-known example of such systems is Google. The main problem with these systems is that they do not have the ability to understand the meanings of the keywords (i.e. semantics). Furthermore, different documents containing same information may be represented differently that makes it more difficult to understand the semantics of the keywords. For example synonyms one of these issues. A synonym is a word that means the same as another word, for example, the company is a synonym of a firm [27].

There are three basic processes an information retrieval system must support: the representation of the content of the documents, the representation of the user's information need, and the comparison of the two representations. Figure 2.5 represents these three processes. Where squared boxes represent data and rounded boxes represent processes [52].

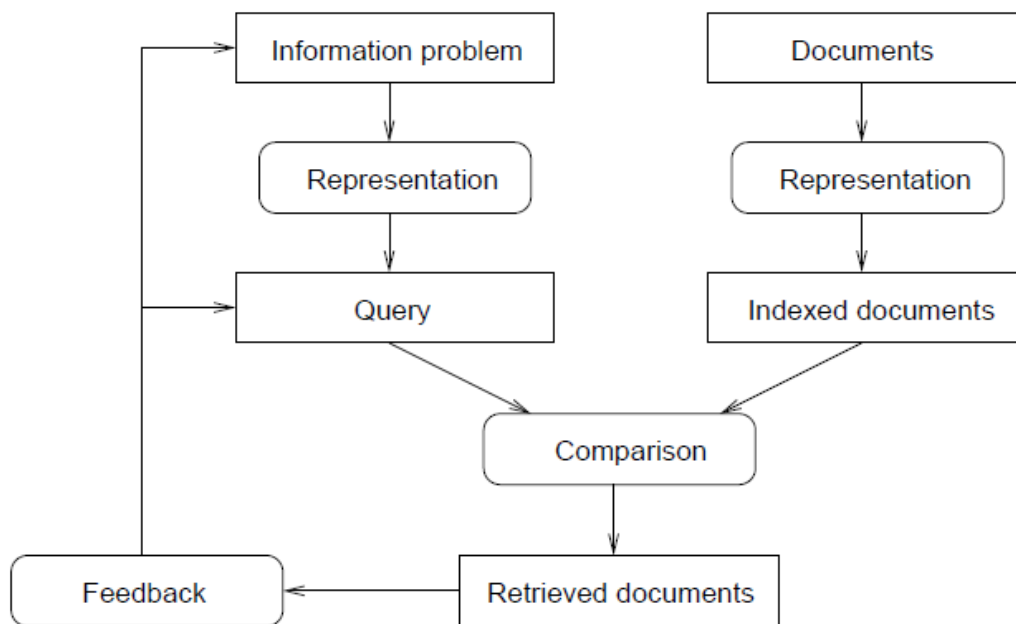


Figure 2.5: Information Retrieval Processes [52]

From Figure 2.5 representation of the documents is usually called indexing process. The process of representing the information problem is usually referred to as the query process. The comparison of the query with the document representations or indexed documents is called the matching process. The matching process results in a ranked list of relevant documents. Over these documents the user can search this documents list about the information he needs. Ranked retrieval will put the relevant documents at the top of the ranked list this will minimize the time the user has to invest in reading the documents.

2.5 Semantic Search Engines

Semantic Search Engines differ from traditional search engines, a semantic search engine stores semantic information about Web resources and is able to solve difficult queries. Semantic search integrates the technologies of the Semantic Web and search engine to enhance the search results obtained from current search engines and evolves to the next generation of search engines built on the Semantic Web. In general, processes of semantic search engine are: The user question is interpreted, extracting the relevant concepts from the sentence, and the results are presented to the user.

Figure 2.6 and Figure 2.7 show the difference between the semantic search engine Hakia (www.hakia.com) vs. traditional search engine Dogpile (www.dogpile.com) based on their expected search results. The phrase that is used to make the comparison is "what is the weather in Kuala Lumpur".

Figures below displayed that Hakia knows what the user want. It is clear from the result displayed by Hakia the information of Kuala Lumpur weather not the website that contain the keyword of Kuala Lumpur weather like Dogpile [28].

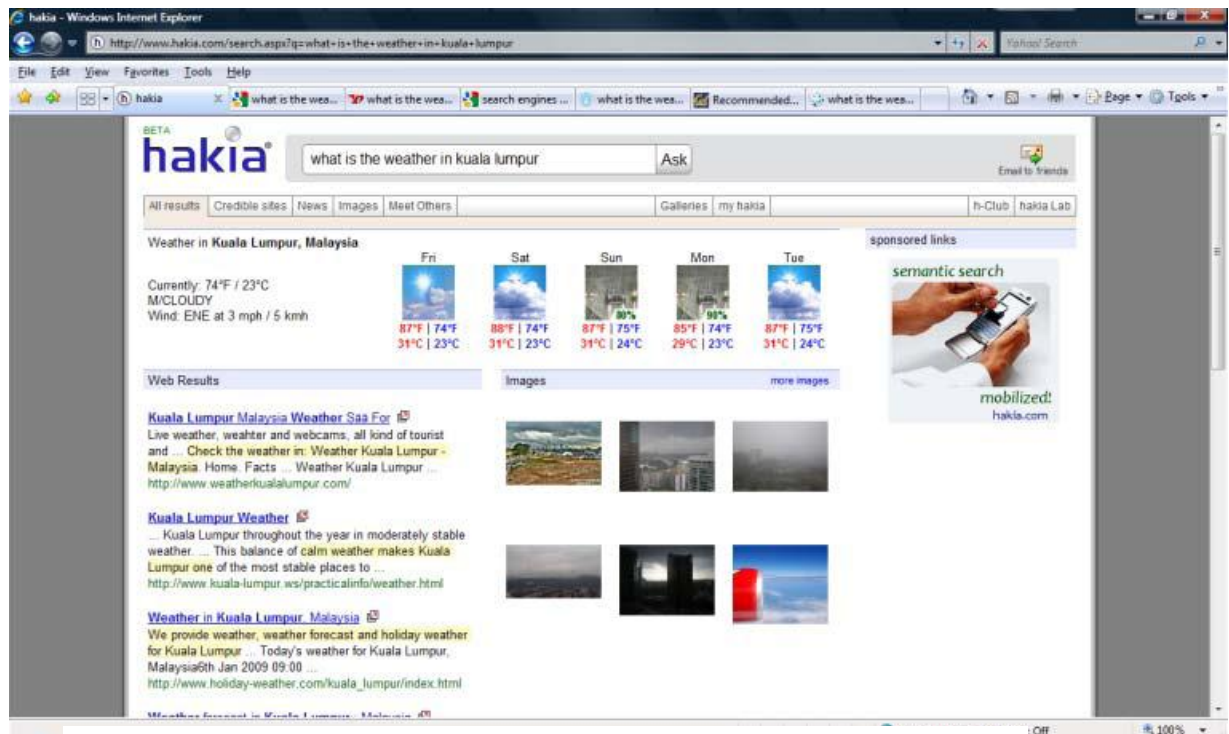


Figure 2.6: Hakia [28]

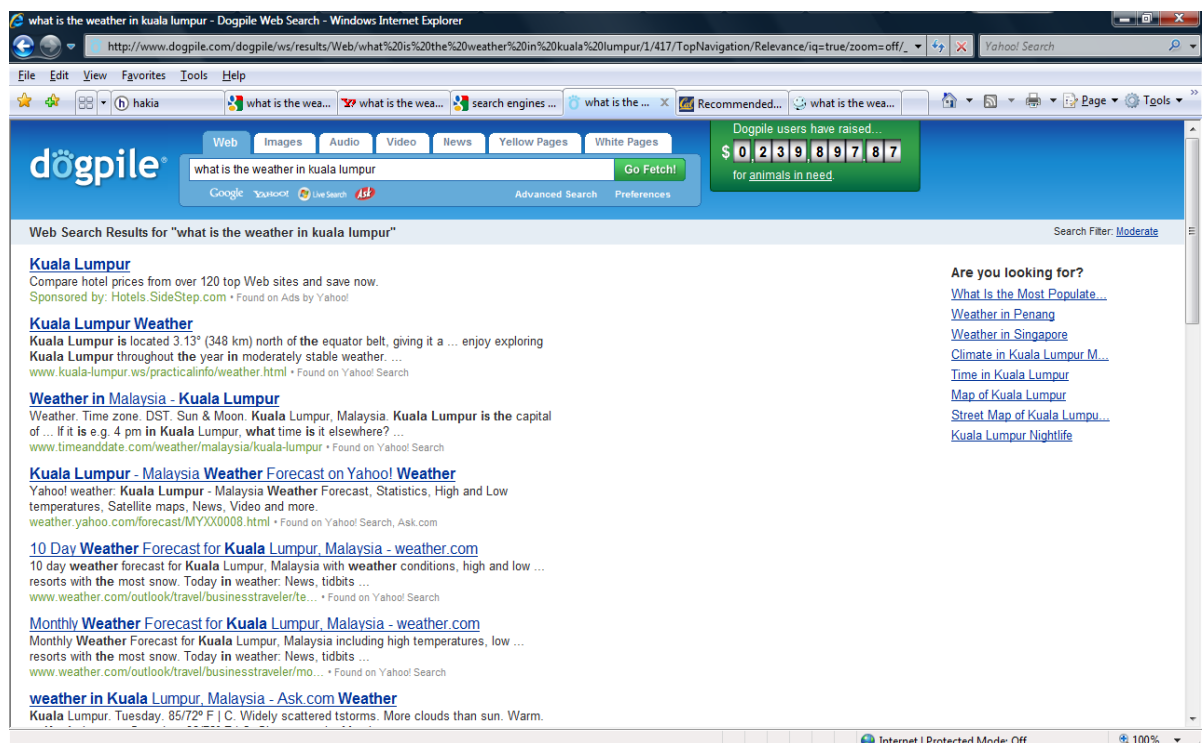


Figure 2.7: Dogpile [28]

From the comparison, a semantic search engine is an answer to overcome the lack of traditional search engine. It is not like a traditional search engine which searches based on keyword, semantic search engine try to analyze and understand the user needs by doing logical reasoning. The result will be more precise.

2.5.1 Semantic Search Engines Examples

- **Hakia**

Hakia is a Web search engine that is concentrated on bringing quality results in all aspects including Web, News, Blogs, Hakia Galleries, Credible Sources, Video, and Images. Some aspects are processed by Hakia's proprietary core semantic technology called QDEXing (Deep Semantics) while others are processed by Hakia's Semantic Rank technology using third party API feeds (Surface Semantics) [50].

- **Go Web**

Internet semantic search engines depends on ontological background knowledge. They use combination of text mining and ontologies to facilitate and enhance question answering biomedical domain. It offers an efficient search and result set filtering mechanism, highlighting and semi-automatic [29].

- **Swoogle**

Is a crawler-based indexing and retrieval system for Semantic Web documents (i.e, RDF or OWL). It analyzes discovered document, and computes relations between documents. Discovered documents are also indexed by an information retrieval system which can use either character N-Gram or URI as keywords to find relevant documents and to compute the similarity among a set of documents. One of the interesting properties computed for each Semantic Web document is its rank - a measure of the document's importance of the Semantic Web [30].

2.6 Summary

In this chapter, we have presented a background for this research. We discussed the concept of Semantic Web and its technologies including ontology, RDF, OWL and the importance of it in the field of information retrieval and search. We presented the semantic annotation and how we can use ontology in the process of annotation. We explained the semantic annotation tools and the difference between them. Additionally we identified the terminology of information retrieval. We also explained the semantic search engines and the difference between a traditional search engine and a semantic search engine by giving an example. The next chapter will present the proposed model used in our work.

Chapter 3: Related Work

In this chapter, different related works are studied and investigated. The related works are introduced and analyzed for using ontology in the process of document annotation, using ontology in the process of information retrieval. Most of presented related works in English language because for Arabic language there is still no published work in this field. Next we present a number of works that are related to ontology-based annotations and retrieval , and ontology-based search and retrieval.

3.1 Ontology-Based Annotations and Retrieval

GoNTogle [6] is a tool that supports manual as well as automatic annotation of different types of documents such as (doc, pdf, rtf, txt). It depends on ontology to achieve this aim. The tool supports different kinds of searching for example Keyword-based search, Semantic-based search. The GoNTogle architecture contains four components: a) Semantic Annotation component which provides some facilities include 3 modules for document viewer, and ontology viewer and an annotation editor, b) ontology server component used for storing semantic annotation of document in the form of owl ontology instances, c) Indexing component Used for indexing documents using an inverted index d) search component that allow users to search for document using both textual (keyword search) and semantic (ontology search) information, The work suffer from limitations the authors depends only the classes and sub-classes of the ontology without depends on properties in the Process of documents annotation.

In [12] they proposed Knowledge and Information Management Platform (KIM), how used KIM in the process of semantic Information retrieval. Also the paper explains using KIM in physical search, pre-defined pattern search and key word search, in the end the paper took about shortcoming in using KIM and future direction to overcome this shortcoming, The work suffers from limitations the authors depends on a specified domain of ontology and we can't make change for this ontology.

In [16] they propose an ontology-based approach to knowledge and document annotation and management (ODCA). The method used to make necessary decisions when specific situations are found and suitable solutions can be searched. The method depends on

structural health monitoring (SHM) ontology which includes nearly 100 classes, 60 object properties and 180 instances; also it can support different search options such as search by queries and full text search. The limitations of the method that it is not showing the tools used, also the architecture not clear.

In [54] they propose Apolda (Automated Processing of Ontologies with Lexical Denotations for Annotation) which consider a freely available plugin for GATE. The Apolda processing resource (PR) annotates a document like a gazetteer, but takes the terms from an (OWL) ontology rather than from a list. Apolda searches the document for OWL annotation properties of the classes and instances of the ontology. The matches are annotated with the name of the class of the ontology. Apolda can be specified by the initialization parameters that used for annotation the documents these parameters include ontology, prefRepresentation, altRepresentation and language where only the ontology parameter consider obligatory.

The limitations of the Apolda annotator it doesn't contain morphological analyzer parameter and Postagger parameter.

3.2 Ontology-based Search and Retrieval

In [17] a method is proposed for Extensible Business Reporting Language (XBRL) database Information retrieval in the field of business, and financial data, the method contains two parts the first part: include XBRL conversation Framework this part concentrate on converting taxonomy which contains all the structural information and definition of metadata into OWL, and XBRL instance documents convert to RDF as the instance of the OWL class. OWL and RDF combine the XBRL database ontology. The second part includes semantic retrieval model this model depends on the XBRL ontology database to retrieve documents. There are a lot of limitations for this work the ontology database doesn't cover the entire contents of economics business and the other limitation that The tools used in the method not clear.

In [18] an Ontology-based Enterprise IR Model proposed for retrieving documents depends on the electric products ontology. The Model contains five components: user interface module where the user can input query in the form of natural language, query processing module which responsible for processing the query by removing as example stop words, the Jena inference engine which will load OWL ontology to deal with the query and is in response to the user query request , query module which responsible about the results that will presented to the user and ontology resource module. To apply the five modules the authors design an algorithm depends on the electric products ontology. The limitations of this work the authors don't show the performance of this method.

In [19] an online Semantic Information Retrieval using Ontologies (SIRO) proposed for retrieve documents, the architecture of the system include three main modules: query processing and enrichment, search and document processing and a module for service classification, the system used for retrieving documents in the web depends on specified query from the user. The main advantages of the system are the use of two ontologies; domain ontology and service ontology improve the relevance of the returned results. Additionally is online system.

In [13] they develop a semantic retrieval system for corn plant ontology, the system contain four modules : ontology building module where they depends on clear strategy to building the ontology which include Document preprocessing, Extraction of feature words, Extraction of semantic triples, user question processing module, document information preprocessing module and query semantic expansion module, the authors in their experiments depends on 100 documents(68 documents is relevant to corn domain knowledge and 32 documents is irrelevant) as experimental dataset the experiments shows that the result comparison with keyword-based retrieval method is better than in precision ratio and recall ratio. This work has some limitations that they build their ontology depends on RDF triple not OWL comparison between RDF and OWL the second consider more expression than the first.

In [14] ontology enabled Web-based multilingual tool for information retrieval in the legal domain is presented, the authors aim of this approach to improve the precision and recall of the search. To build ontology, and to retrieve documents they used Protégé and its query engine, they used the approach to retrieve documents written in Arabic, the retrieval process is also enriched by enabling the user to retrieve English or French documents, in order to do that the original query is translated using machine translation to the target language French or

English, they use wordnet to extend the translated query, the limitations of the method it not clear how the process of search execute.

In [15] they present ontology based information retrieval system which depends on sport ontology and SPARQL query language to retrieve documents, the system searches sports information by the semantic relationships between concepts defined in the ontology according to relations of “Synonym of”, “kind of” and “part of” between sports concepts.

The process of ontology-based information retrieval includes multiple steps begin by creating a domain ontology, collect a dataset from the sources to annotate dataset using an ontology, the search engine used to complete semantic matching of retrieval conditions over ontology reasoning to find out the correct dataset, the last step includes the results which back to the user. The limitations of the system that it is difficult for normal user to make query depends on SPARQL query this mean we need professional users.

3.3 Summary

In this chapter we presented an overview about some of researches conducted in documents annotation and retrieval, were presented the documents annotation and retrieval methods based on an ontology as an play vital role in the process of document annotation. Most of the work presented in the English language this mean there is no work until now for Arabic language. We preferred to work at the documents annotation and retrieval in Arabic, because it is a new and try to solve the problem in this field.

Chapter 4: The Proposed Model

In this chapter, we develop an automatic ontology-based document annotation and retrieval model for Arabic documents. Our model will be used to improve the accuracy of Arabic retrieved documents depending on Arabic Ontology Domain "فقہ الصلاة" (Prayer jurisprudence). All Documents in this domain are written in Arabic language and stored in a corpus.

To build the model various steps have to be performed based on our methodology (see Section 1.5). The main required steps are shown in Figure 4.1 and stated below:

- Preparing the corpus
- Building Arabic Ontology Domain "فقہ الصلاة" (Prayer jurisprudence)
- Documents annotation
- Processing annotated documents
- Indexing and searching

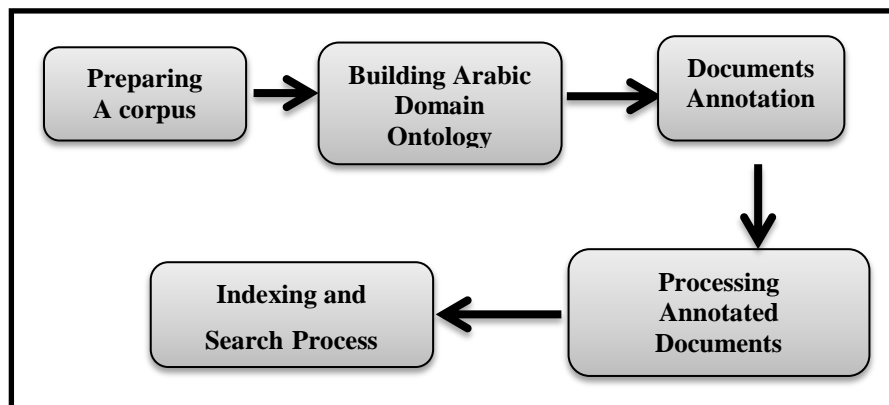


Figure 4.1: Execution Steps

4.1 Preparing the Corpus

Preparing the corpus is one of the most important stages in the research project. The corpus is a collection of documents in one domain. We use these documents in the process of annotation and retrieval. In our work we collect nearly 100 documents related to our Arabic Ontology Domain "فقہ الصلاة" (Prayer Jurisprudence). We collect these documents from Islam

Web [37] website related to Fatwa questions in the field of Islam issues. The website contains a comprehensive opinions of Islamic scholars related to daily issues of Islam. We concentrate in the part of "فقه الصلاة" (Prayer Jurisprudence) and build our ontology depending on it. All documents that are gathered from the website have direct relation with ontology components. All the documents collected is converted to xml type when we load it into Gate in order to facilities the processing of documents annotation and retrieval.

4.2 Ontology Building

Building ontology is an important task in our work, we used a top-down approach [38] in building the ontology. Most abstract concepts are identified first, then specialized into more specific concepts to build our Arabic Ontology Domain "فقه الصلاة" (Prayer Jurisprudence) which represents the basic knowledge in our work. We construct the ontology manually by helping experts in the field of "فقه الصلاة" (Prayer Jurisprudence) which is the source for the main components of the ontology.

The development of ontology consists of the following stages:

- Define concepts, i.e., classes based on studying and analyzing the domain.
- Define instances, i.e., real elements in our domain.
- Define relations among classes as a requirement to come up with the ontology.
- Enrich ontology with Synonyms and Stemming words.
- Ontology Evaluation

Stage 1: Define concepts, i.e., classes and sub-classes

The concepts include classes, sub-classes as an example used in our ontology domain. These concepts are not selected randomly but are selected depending on our corpus. This means we concentrate on all issues that interest users related to our domain "فقه الصلاة" (Prayer Jurisprudence).

The selected concepts depend on some questions related to the ontology including what is the expected concepts used from users when they search over corpus?

How is the relations between these concepts represented?

Our ontology contain 23 concepts including classes, sub-classes. Table 4.1 Depicts the selected ontology concepts.

Table 4.1: Ontology Classes

No.	Classes-Subclasses /Arabic	Classes-Subclasses /English	Description
1	وقت الصلاة	Prayer Time	The time of FardhuAin prayer
2	الأذان	Aladan	Aladan is the call to prayer itself, and the person who calls it is called the muadhan.
3	السهو	Omission	Forget one of the prayer steps
4	سهو زيادة	Increase Omission	Either increase in acts or statements when the person does the prayer
5	سهو شك	Omission Doubt	Doubt between the two things, whichever is signed throughout the prayer
6	سهو نقصان	Decrease Omission	Either increase in acts or statements when the person does the prayer
7	شروط الصلاة	Prayer Conditions	Matters that are not part of the prayer, but must be satisfied before starting the prayer
8	شروط صحة	Validity Conditions	Conditions of prayer being valid refer to that on which the validity of prayer depends, such that if one of these conditions is broken, then prayer is not valid as a result.
9	شروط وجوب	Obligation Conditions	Conditions of prayer must be available in the person who want to pray to be his prayer right.
10	صلاة التطوع	Voluntary Prayer	It is the optional prayer can do beside the obligatory prayer
11	سنن رواتب	AlRoateb Sunan	Beyond the five daily required prayers, Muslims often engage in optional prayers before or after the regular prayers (FardhAin). These are known as "AlRoateb Sunan".
12	رواتب بعدية	Post-Roateb	It is done after the FardhuAin prayer
13	رواتب قبلية	Pre-Roateb	It is done before the FardhuAin prayer
14	صلاة العيد	Eid Prayer	Eid prayer is performed on the morning of Eid ul-Fitr and Eid ul-Adha.

15	صلاة أهل الأعذار	Prayer of Exempted People	Persons who have a problem which can't do the prayer in suitable way.
16	صلاة فرض	Obligatory Prayer	The prayer must done by every person
17	فرض عين	FardhuAin	It is the main five prayers that done by person who want to pray.
18	فرض كفاية	FardhuKifayah	Prayer that carried out by one fall for others
19	مكونات الصلاة	Prayer Components	The main components for prayer and must be found in it include (Staff, Disliked, things which invalidate and Musthbat).
20	أركان	Staff	It is one of the important components of prayer related with the practical side.
21	مكروهات	Disliked	Things that are unlike in prayer
22	مبطلات	Things which Invalidate	Things make prayer wrong
23	مستحبات	Musthbat	Things that are preferred in the prayer

In Table 4.1 contains the 23 ontology concept mentioned in our domain "فقه الصلاة" (Prayer Jurisprudence). Choosing these concepts has direct relation with user requirements used in the process of search. We mention every ontology concept in Arabic and its synonym in English including the description of the concept. Some of these concepts have relations with other concepts and this helps in the search process to retrieve more results. Also, most of these concepts have synonym words and they contain instances to help in the process of documents annotation.

Figure 4.2 depicts part of ontology concepts used, in the figure, for example, class "فقه الصلاة" (Prayer Jurisprudence) considers the root class and other classes branches from it. Also we can find the class "شروط الصلاة" (Validity-Conditions) consider sub-class from the class "شروط الصلاة" (Prayer-Conditions).

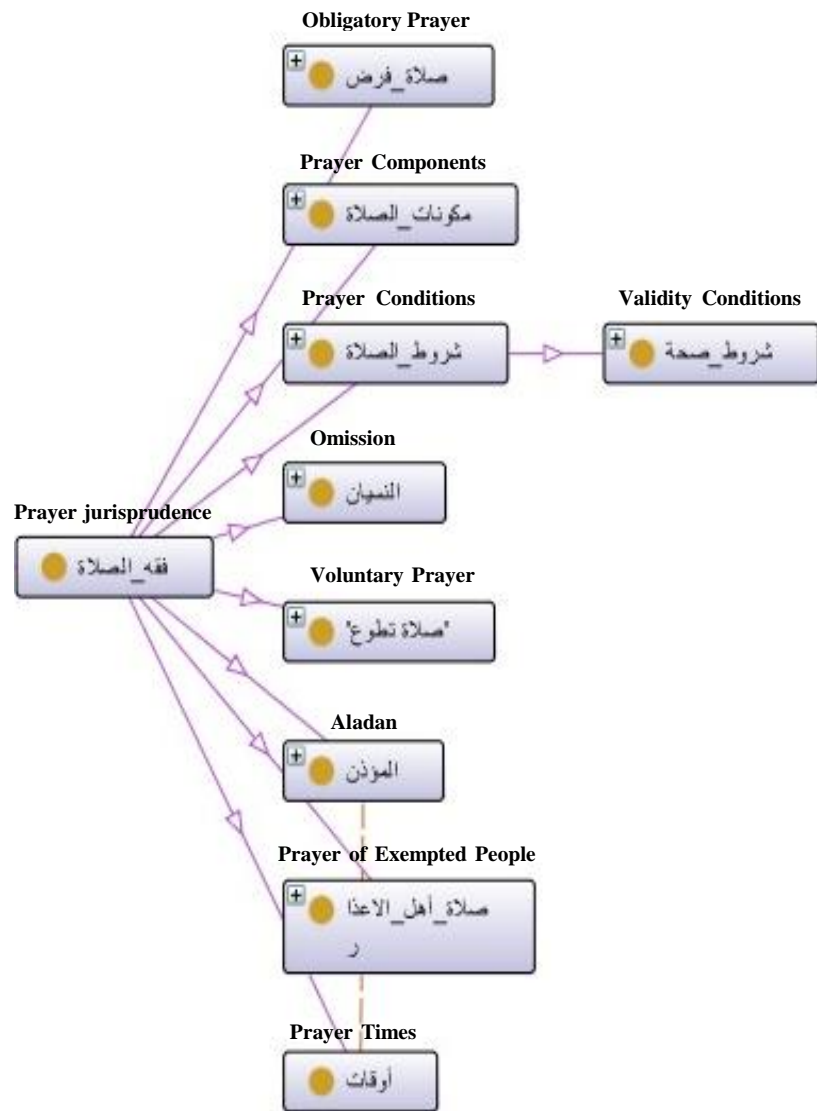


Figure 4.2: Part of the Ontology Concepts

Figure 4.3 depicts part of the concepts and instances used in the ontology domain "فقه الصلاة" (Prayer Jurisprudence). In the figure, for example, the class "فرض عين" (FardhuAin) have several instances include ("العصر" (Asr), "المغرب" (Maghrib), "العشاء" (Isha), "الظهر" (Duhr), "الفجر" (Fajr)). Also the class "صلاة أهل الأعذار" (Prayer of Exempted People) have several instances include ("الخائف" (Afraid), "الراكب" (Passenger), "المريض" (Patient), "المسافر" (Traveller)).

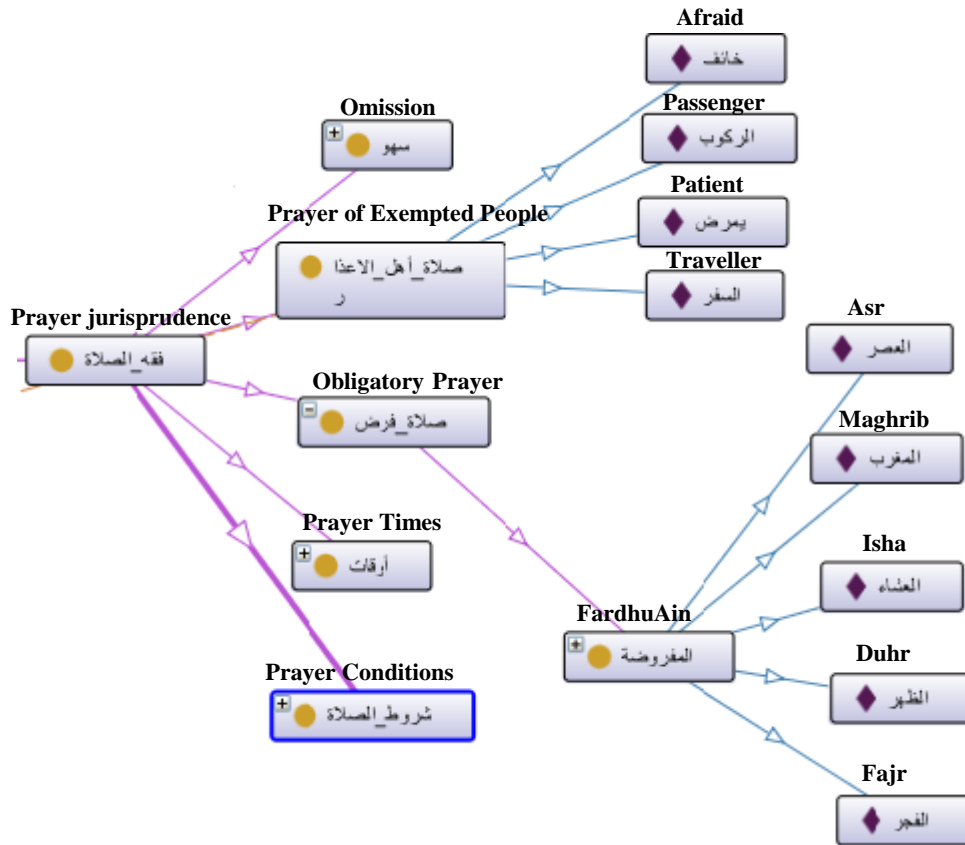


Figure 4.3: Part of Ontology Concepts and Instances

Stage 2: Define instances, i.e., real elements in the chosen domain.

Creating instances (individuals) is a very important step to enrich the ontology with direct relation with classes and sub-classes.

In our ontology we defined around 58 instances representing all ontology concepts. Figure 4.4 depicts some of these instances.

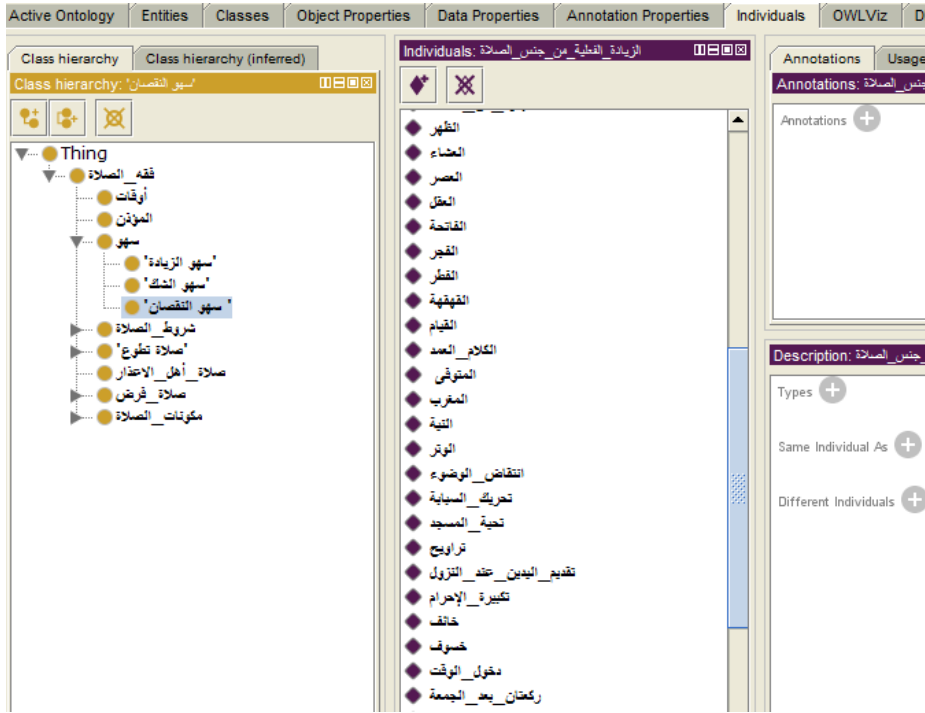


Figure 4.4: Ontology Instances

Stage 3: Define object properties (relations) among classes as a requirement to come up with the ontology. Create object properties play a vital role in connecting concepts of the ontology in our Arabic Ontology Domain "فقه الصلاة" (Prayer jurisprudence).

We used 4 object properties that connect the important concepts which have relations with each other. The main cause for using only 4 object properties is due to the nature of the ontology where there is no suitable relations between ontology components.

Table 4.2 depicts these 4 object properties in our Arabic ontology.

Table 4.2: Ontology Object Properties

No.	Object property/ AR	Object property /EN
1	تخفف_فيها	Alleviate
2	مرتبطة_بصلاة	Linked to Prayer
3	لا_ترتبط	Not Linked
4	يؤذن	Inform

Stage 4: Enrich ontology with synonyms and stemming

In our work we don't need any stemmer for stemming documents or Gazetteer list for synonyms words. We solve these two issues by enriching our ontology with a lot of synonym words and stemming words, where synonym is a word that means the same meaning or similar meanings for another word [51], and stemming is the process for reducing or sometimes derived words to their stem, base or root form [39]. For example the synonyms for instance "الجنابة" (Funeral) is "الميت" (Dead) and "المتوفى" (Dead) and the stemming words for instance "المسافر" (Traveler) is "السفر" Traveling, "يسافر" (Travel), "مسافر" (Traveler), and "سفر" (Travel). Figure 4.5 depicts an example of the synonyms words for instance "الجنابة" (Funeral) which is "الميت" (Dead) and "المتوفى" (Dead) in our Arabic ontology domain. We express synonyms and stemming by using the annotation property: Label when we build the ontology using protégé tool which is represented as `rdfs:Label` in the ontology. Figure 4.5 depicts the synonyms words for Instance "الجنابة" (Funeral).

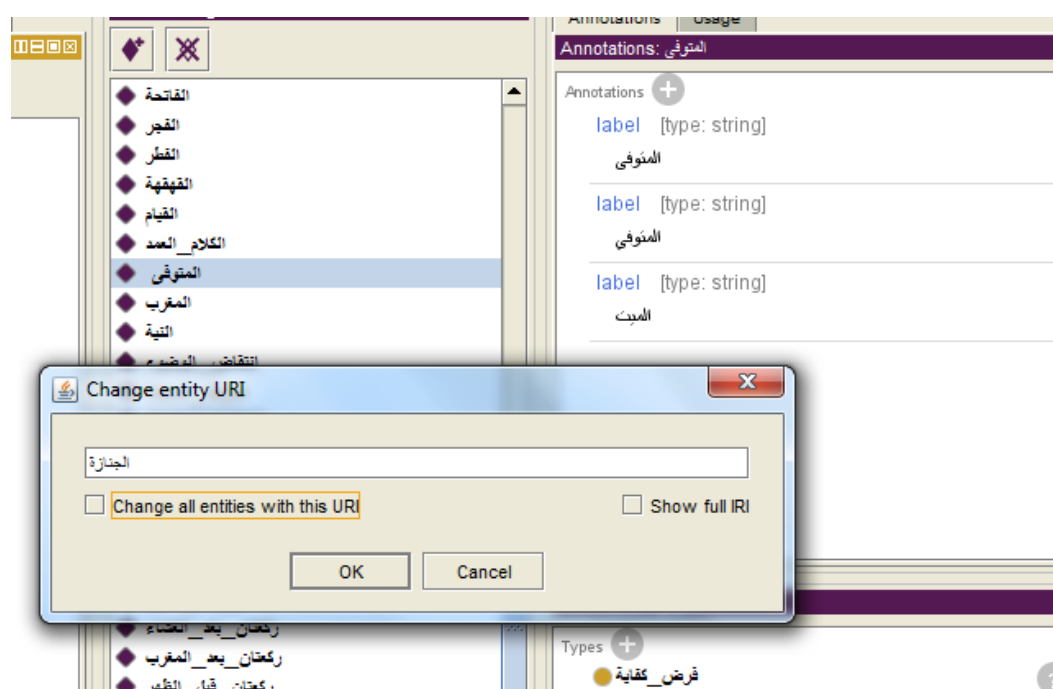


Figure 4.5: The Synonyms Words for Instance "الجنابة" (Funeral)

Stage 5: Ontology evaluation

The important thing after building the ontology is the process of evaluating it. We depend on the Golden standard and the help of domain experts. We use precision and recall [45] to evaluate the ontology.

Correct concepts are decided based on the Golden standard which could be another ontology, or it could be taken statistically from a corpus of documents or prepared by domain experts. Precision is number of correct concepts in the ontology relative to the total number of concepts in the ontology as shown in equation 4.1 and Recall is number of correct concepts in the ontology relative to total number of possible concepts as shown in equation 4.2.

$$\text{Precision} = \frac{\text{Number of correct concepts in the ontology}}{\text{Total number of concepts in the ontology}} \quad \dots \text{ eq (4.1)}$$

$$\text{Recall} = \frac{\text{Number of correct concepts in the ontology}}{\text{Total number of possible concepts}} \quad \dots \text{ eq (4.2)}$$

In general precision is a metric that is used to indicate how accurately the concepts identified in the ontology represent the domain. Recall is used to measure the coverage of the ontology.

In our case we relied on the domain expert to evaluate the ontology by asking him about the shortage in the ontology concepts/classes. He identified 23 correct classes and the total number of classes is 23 then the Precision would be:

$$\text{Precision} = 23/23=100\%$$

Also in our case the domain expert said there is still missing 1 concept/class that the ontology does not cover. Then the total number of possible concepts equal 24 and the Recall would be:

$$\text{Recall} = 23/24= 95.83\%$$

The same way we can calculate the instances or individuals of the ontology by asking the domain expert about the shortage in ontology instances the domain expert identified 58 correct instances, therefore the Precision would be:

$$\text{Precision}=58/58=100\%$$

Also in our case the domain expert said there is still missing 12 instances or individuals that the ontology does not cover. Then the total number of possible instances equal 70 and the Recall is:

$$\text{Recall} = 58/70 = 82.85\%$$

Table 4.3:Ontology Evaluation

Metric	Classes	Instances (Individuals)
Precision	100%	100%
Recall	95.83%	82.85%

4.3 Documents Annotation

Semantic annotation is performed on the xml documents using the Onto Root Gazetteer Annotator which is in combination with few other generic Gate resources capable of producing ontology-based annotations over the given content with regards to the given ontology. This gazetteer is a part of Gazetteer_Ontology_Based plugin [40]. The parameters for Onto Root Gazetteer include:

- **Morpher**: is the identification, analysis and description of the structure of a given language.
- **PosTagger**: produces a part-of-speech tag as an annotation on each word or symbol.
- **Ontology**
- **Tokenizer**

In our work we only are interested in the last two parameters; ontology and tokenizer because at this stage of our work we don't need Morpher and PosTagger. They are needed as parameters in order for Onto Root Gazetteer to work properly.

As we said the ontology is considered the main important phase which we use in the process of documents annotation. In this phase all the documents that are saved in the corpus and contain words and have relation with the ontology components will annotate with these ontology components and the result will be annotation types (ontology classes). Figure 4.6 depicts using Onto Root Gazetteer annotator. It includes two parts in the right side we can watch Onto Root Gazetteer annotator loaded and on the left side the parameters used for Onto Root Gazetteer also loaded in the bottom side of the figure we can select the corpus for our work.

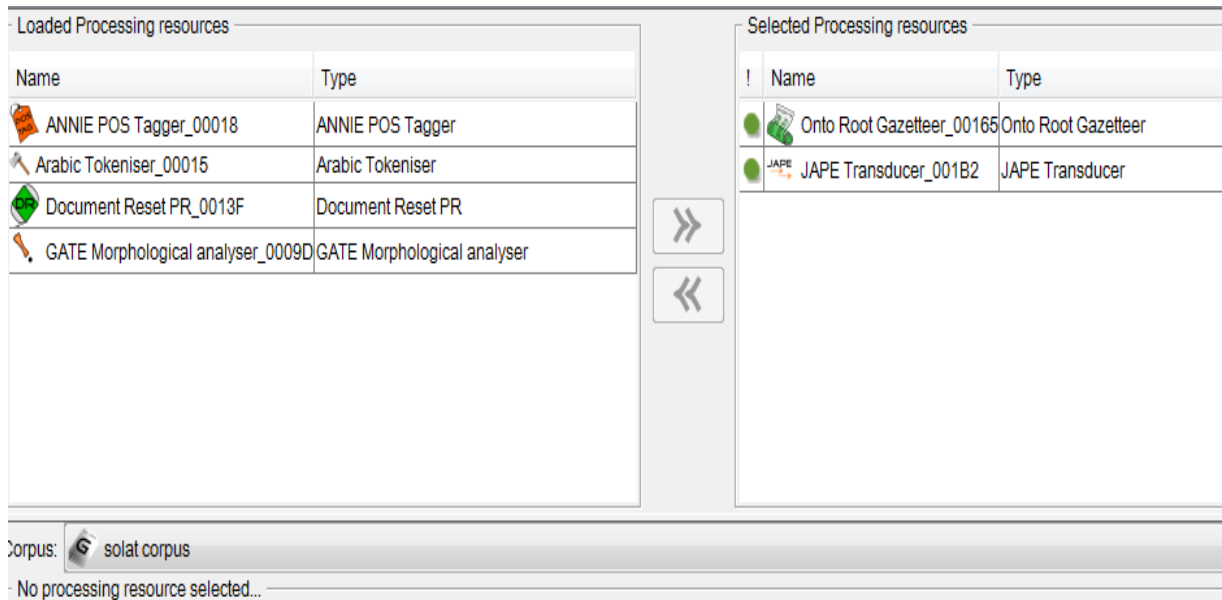


Figure 4.6: Using Onto Root Gazetteer

Figure 4.7 depicts a sample of the result using Onto Root Gazetteer annotator, the example shows the annotation types "صلاة أهل الأعدار" (Prayer of Exempted People) and "رواتب" (Roateb) in the figure when we select the two annotation types the ontology components that have related with words in the text will annotate.

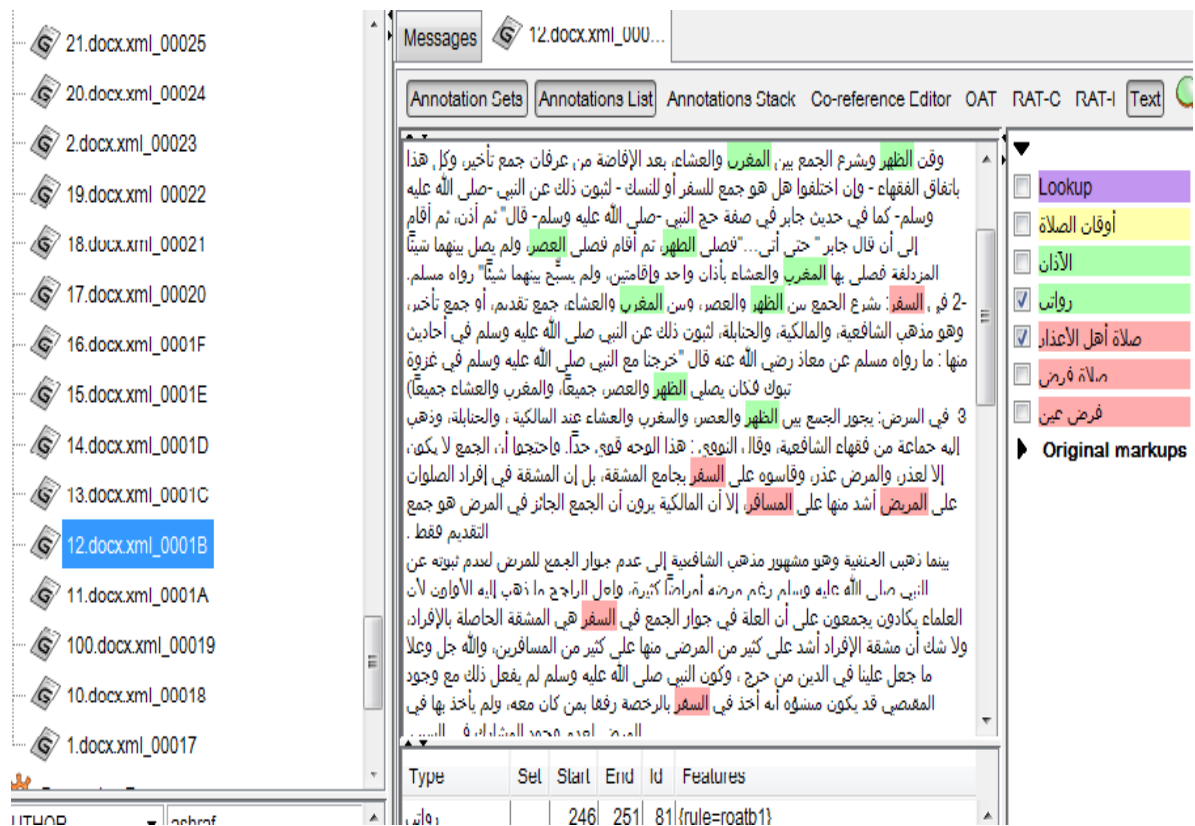


Figure 4.7: The Annotation Process Result

- **Tokenizer**

Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. The list of tokens becomes input for further processing such as parsing or text mining. Tokenization is useful both in linguistics (where it is a form of text segmentation), and in computer science, where it forms part of lexical analysis [41]. Gate have a lot of tokenizers for different languages including Arabic tokenizer. We tokenize the documents into tokens this facilitate the process of documents annotations. The Arabic tokenizer will be the important parameter for Onto Root Gazetteer beside the ontology. The Arabic tokenizer will splits every document stored in the corpus into very simple tokens such as numbers, words of different types this help in the process of annotations then these tokens will annotate with ontology components such as classes, sub-classes, relations where the ontology consider the other parameter for Onto Root Gazetteer annotator.

4.4 Processing Annotated Documents

In this step after annotating documents using Onto Root Gazetteer annotator, we pass annotated documents to Jape Transducer plugin [53] which has init-time parameter Grammar URL, that appears as an optional parameter to the grammar URL. The User can use this parameter (i.e. Grammar URL) to specify the Jape rules [42] that consider files written with the extension ".jape", the Jape Transducer parse and compile the Jape rules at run-time to execute them over the GATE document(s). The output of processing these documents will be the annotation types (ontology classes) for example "صلاة أهل الأعدار" (Prayer of Exempted People), "رواتب" (Roateb) and "فرض عين" (FardhuAin). All Jape rules used in our work rely on the "annotation type lookup" which is the default annotation type that we use as input for all Jape rules used in our work. Figure 4.8 depicts some of Jape rules used in processing annotating documents.

```

}
Rule: salataheladr1
(
{Lookup.URI=="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الخائف"}
):label
-->
{
gate.AnnotationSet label = (gate.AnnotationSet)bindings.get("label");
gate.Annotation personAnn = (gate.Annotation)label.iterator().next();
gate.FeatureMap features = Factory.newFeatureMap();

features.put("rule", "salataheladr1");
outputAS.add(label.firstChild(), label.lastNode(), "صلاة أهل الأعداء",
features);
}

Rule: salataheladr1
(
{Lookup.URI=="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الراكب"}
):label
-->
{
gate.AnnotationSet label = (gate.AnnotationSet)bindings.get("label");
gate.Annotation personAnn = (gate.Annotation)label.iterator().next();
gate.FeatureMap features = Factory.newFeatureMap();

features.put("rule", "salataheladr1");
outputAS.add(label.firstChild(), label.lastNode(), "صلاة أهل الأعداء",
features);
}

```

Figure 4.8: Jape Rules Example

for example the instances "الخائف" (afraid) and "الراكب" (Passenger) will be under the annotation type "صلاة أهل الأعداء" (Prayer of Exempted People), so we need only the annotation type "صلاة أهل الأعداء" (Prayer of Exempted People) to use it in the process of search that will return all documents contain the ontology instances "الخائف" (afraid) and "الراكب" (Passenger) that will be under it.

4.5 Indexing and the Search Process

The purpose of indexing is to optimize speed and performance in finding relevant documents for a search query. Without an index, the search engine would scan every document in the corpus, which would require considerable time and computing power. For example, while an index of 10,000 documents can be queried within milliseconds, a sequential scan of every word in 10,000 large documents could take hours [43]. The corpus of all annotated documents will be stored in Datastore of Gate to begin the process of indexing and searching. We depend on Lucene Datastore search engine in this process which is part of Gate environment.

4.6 The Model Structure

Figure 4.9 provides a general view of the architecture of our model.

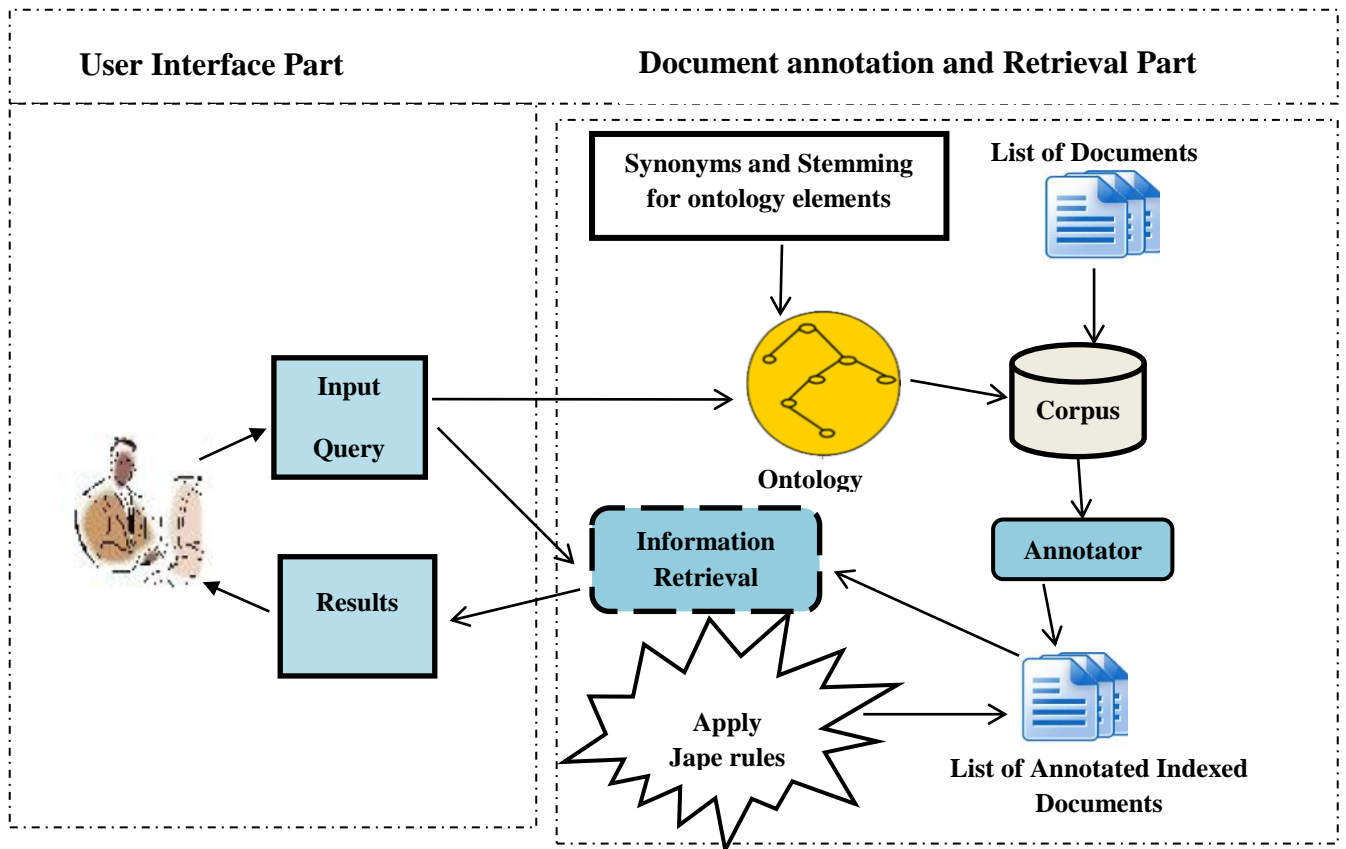


Figure 4.9: The Proposed Model

The proposed model contains the following components:

- **User Interface:** is used by the user to input the query as annotation type (ontology class) and view the results of the retrieved documents. It uses the following components:
 - **Input Query:** The user inputs the query as annotation type (ontology class) in Arabic and the Lucene search engine make the search process.
 - **Results:** The system shows the relevant documents that have a relation with the query input.
- **Document annotation and retrieval process:** Is performed in order to retrieve the relevant documents by entering the query over the user. It consists of the following components:
 - **Annotator:** Used to make the annotations for annotated documents in the corpus.
 - **Jape rules:** then Jape rules apply in the annotated documents.

- **Information retrieval process:** Used to find the relevant document to appear as a result of the user request.
- **Ontology:** Domain specific classes, sub-classes, and properties the ontology contain synonyms and stemming words.
- **Corpus:** this is a repository for documents that will be annotated and used in the process of information retrieval.

From the above (specifically, document annotation and retrieval), after all components interact and execute together the result will be annotated documents. The user can begin writing his query in Gate interface which will be the annotation type (ontology class) that come from the process of documents annotation and then send the query. The search process starts over the annotated documents and the results will be presented in the Gate interface in the form of documents that match the query.

4.7 Summary

In this chapter, we have discussed the steps to execute our methodology. In the first section, we talked about preparing our corpus of documents the number of documents and how we obtain it. In the second section, ontology building stages explained. In the third section documents annotation steps explained depending on Onto Root Gazetteer Plugin. In the fourth section we explain how we processing annotated documents using Jape Rules. In the fifth section the indexing and search process explained which execute depending on the Lucene search engine. The next chapter will present the implementation, experimental results and evaluation.

Chapter 5: System Realization, Experimental Results and Evaluation

This chapter presents the implementation and the evaluation of our work. Firstly we state the tools and programs used to develop the proposed model, the system interface, test the system and evaluate its performance is explained next. At the end of the chapter we shall discuss our results.

5.1 System Realization

We base our realization on the model structure presented on Section 4.6 (The Model Structure).

5.1.1 Tools and Programs

To compose our system, we utilize the following tools and programs.

- For indexing and keyword searching we use Lucene Datastore search engine.
- Protégé for ontology building.
- Gate as environment to execute all our work.

5.2 System Interface

The main Gate interface is shown in Figure 5.1 and consists of the following parts:

1. **Applications:** in this part we execute our application which we name it "تطبيق الصلاة" (Prayer Application), by adding the plugins and Jape rules in its pipeline.
2. **Language Resources (LRs):** represent entities such as lexicons, corpora or ontologies.
3. **Processing Resources (PRs):** represent entities that are primarily algorithmic such as parsers.
4. **Data stores:** specialized folder on a hard drive used to store the annotated corpus and improve processing times for large collections of documents.
5. **Text area:** view the document before and after the annotation.

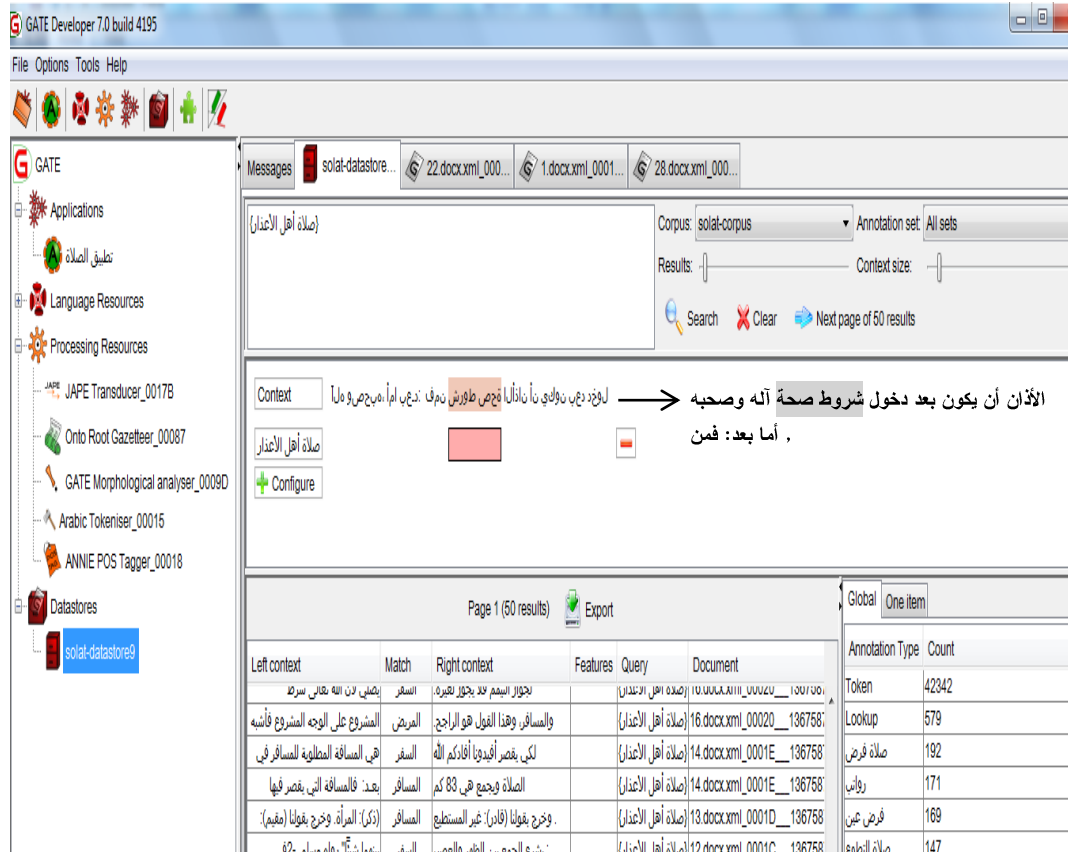


Figure 5.1: Gate Interface

To realize our prototype, we follow a number of steps after building the ontology and preparing the corpus (see Section 5.2).

1. Language Resources (LRs)

- Loading a corpus which contains documents related to our Arabic Ontology Domain "فقہ الصلاة" (Prayer jurisprudence), then we will convert these documents to xml type to help us in the process of documents annotation and retrieval. We load documents to Gate as Unicode (UTF-8) to support Arabic language.
- Loading the Arabic Ontology Domain "فقہ الصلاة" (Prayer jurisprudence) that build using Protégé GATE program and imported by Gate.

2. Processing Resources (PRs)

- Loading Arabic Tokenizer which is used to tokenize every document stored in the corpus into very simple tokens as mention in Section 4.3, ANNIE POS Tagger and Gate Morphological analyzer Plugins.

- Loading Onto Root Gazetteer which is considered as the Annotator in our work. The Onto Root Gazetteer has multiple parameters. It contains all Plugins mentioned in Section 4.3 including Arabic Tokenizer, ANNIE POS Tagger, Gate Morphological analyzer and the ontology that play the main role in the process of document annotation.
- Loading Jape Transducer Plugin which depends on Jape rules as parameters. These Jape rules are used in processing annotated documents, taking into account the Unicode windows-1256 for Arabic language.

3. Applications

In this part we connect and realize the parts as one running system. As soon as we run the application which we name "تطبيق الصلاة" (Prayer Application), all parts (Plugins) mentioned in (Section 4.3) are loaded automatically. Then we choose Onto Root Gazetteer and Jape Transducer and apply these two parts (Plugins) on the selected corpus. After that we can see the annotated documents.

4. Data stores

In this part we create the Datastore to perform two steps:

- Storing the corpus of annotated documents.
- Do indexing for annotated documents using the Lucene Datastore search engine.

5.3 Experiments

We performed a series of experiments to demonstrate the ability of our system to retrieve the related documents. All our experiments depend on the annotation types (ontology classes) that created from the processing of annotated documents using Jape rules.

We give some examples to demonstrate and test the prototype and search using the annotation types that come up with the process of documents annotation. Figures 5.2 , 5.3 and 5.4 are three examples showing the results of a search using three annotation types "صلاة أهل الأعدار" (Prayer of Exempted People), "رواتب" (Roateb) and "الأذان" (Aladan). Figure 5.5 is one example for using the word "الأذان" (Aladan) as keyword (traditional way) in the search.

Example 1. Searching using annotation type "صلاة أهل الأعدار" (Prayer of Exempted People).

The screenshot shows the GATE software interface. The search query is "صلاة أهل الأعدار". The results table is as follows:

Left context	Match	Right context	Features	Query	Document
الجماعة لا فرقى فيه بين	الخوف	على النفس، والخوف على العالم.		{صلاة أهل الأعدار}	9.docx.xml_00071__136785
وغيرهم كثير أن له حكم	المسافر	من القصر والإفطار، لعموم النصوص،		{صلاة أهل الأعدار}	17.docx.xml_00021__13678
يصلى لأن الله تعالى شرط	السفر	لجوار التيمم فلا يجوز لغيره.		{صلاة أهل الأعدار}	16.docx.xml_00020__13678
كان منكم مريضاً أو على	سفر	فعدة من أيام أخر وعلى		{صلاة أهل الأعدار}	10.docx.xml_00019__13678
لعذر والمرضى عذر وقاسوه على	السفر	بجامع المشقة، بل إن المشقة		{صلاة أهل الأعدار}	12.docx.xml_0001C__13678
المشقة في أفراد الصلوات على	المرضى	أشد منها على المسافرين إلا		{صلاة أهل الأعدار}	12.docx.xml_0001C__13678
على المرضى أشد منها على	المسافر	، إلا أن المالكية يرون أن		{صلاة أهل الأعدار}	12.docx.xml_0001C__13678

The context window shows the sentence: "بالرخصة رفقا بمن كان معه، السفر منشؤه أنه أخذ في". The search results are displayed on Page 1 (60 results).

Figure 5.2: Search Using Annotation Type "صلاة أهل الأعدار" (Prayer of Exempted People)

In Figure 5.2, when we search using the annotation type "صلاة أهل الأعدار" (Prayer of Exempted People) the search process returns different matching's of the words. For example the search process returns "الخوف" (Fear), "المرض" (Disease) and "المسافر" (Traveler). When we look to the word "الخوف" (Fear), we find that it is a from of the stemming words for the word "الخائف"(Afraid) and the word "المرض" (Disease) is a from of the stemming words for the word "المريض" (Patient).

Example 2. Searching using annotation type "رواتب" (Roateb).

The screenshot shows the GATE software interface. The search query is "{رواتب}" and the corpus is "solat-corpus". The search results are displayed in a table with the following columns: Left context, Match, Right context, Features, Query, and Document. The results are sorted by count, with "رواتب" having 171 matches. Other related terms include "المغرب" (169), "المفروضة" (192), "الفجر" (147), "صلاة الطوع" (147), "صلاة العبد" (68), "مكونات الصلاة" (58), and "السهو" (57).

Left context	Match	Right context	Features	Query	Document
ما لم تصفر الشمس" والحديث العصر الله عليه وسلم قال: "وفى				{رواتب}	66.docx.xml_00057_1367851232
، وللإفادة في هذا الموضوع نقول: المفروضة فلا تصح إلا بعد أداء				{رواتب}	21.docx.xml_00026_1367851231
حين صار ظل كل شيء الظهر الصائم، وصل في العرة الثانية				{رواتب}	67.docx.xml_00058_1367851232
، ويصح أدائها بعد الصلاة عند المفروضة وفي الصلاة، وقبل أداء الصلاة				{رواتب}	21.docx.xml_00026_1367851231
وحرم الطعام على الصائم، وصل الفجر ثم صلى الفجر حين برق				{رواتب}	67.docx.xml_00058_1367851232
، وركعتان بعد العشاء، وركعتان قبل المغرب الظهر، وركعتان بعد				{رواتب}	21.docx.xml_00026_1367851231
، وركعتان قبل الفجر، وأما الرواتب العشاء، وركعتان بعد المغرب، وركعتان بعد				{رواتب}	21.docx.xml_00026_1367851231
حين كان الظل مثله، ثم الظهر أسفر قليلاً، ثم صلى به				{رواتب}	65.docx.xml_00056_1367851232
في الأول، منها حين كان الظهر عند البيت مرتين، فصل، بـ				{رواتب}	67.docx.xml_00058_1367851232

Figure 5.3 : Search Using Annotation Type "رواتب" (Roateb)

In Figure 5.3, when we search using the annotation type "رواتب" (Roateb) the search engine returns many results that have direct relation to returned words or indirect relation with returned words. Also the search process returns different matching of words for example when we search using "رواتب" (Roateb) it will return "المغرب" (Maghrib), "المفروضة" (Obligatory), and "الفجر" (Subh). When we look to the word "المفروضة" (Obligatory) it is the synonym word for the word "صلاة فرض" (Obligatory Prayer).

Example 3. Searching using annotation type "الأذان" (Aladan).

The screenshot shows a search interface with the following components:

- Search Bar:** Contains the query "{الأذان}" and a dropdown menu for "Corpus: solat-corpus".
- Results:** A search bar with a magnifying glass icon, a "Clear" button, and a "Next page of 50 results" button.
- Context:** A text area showing the search results in Arabic: "طبعاً الإمام لم يبدأ الخطبة الأذان يجوز الكلام يوم الجمعة أثناء".
- Annotation Type:** A dropdown menu showing "الأذان" and a "Configure" button.
- Table:** A table with 6 columns: Left context, Match, Right context, Features, Query, and Document. The table contains 13 rows of search results. The "الأذان" annotation type is highlighted in the "Annotation Type" column.
- Left Panel:** A list of documents and resources, including "13.docx.xml_00020", "12.docx.xml_0001F", "11.docx.xml_0001E", "100.docx.xml_0001D", "10.docx.xml_0001C", "1.docx.xml_0001B", "solat-ontology.owl_00017", "Processing Resources", "JAPE Transducer_0017E", "Onto Root Gazetteer_0008A", "GATE Morphological analyser_0009D", "Arabic Tokeniser_00015", "ANNIE POS Tagger_00018", "Datastores", and "solatdatastore16".

Figure 5.4: Search Using Annotation Type "الأذان" (Aladan)

In Figure 5.4 when we search using annotation type "الأذان" (Aladan), the search engine returns a mix of many results. The retrieved documents have relations with both annotation types: "الأذان" (Aladan) and "أوقات الصلاة" (Prayer Times). That is because there is direct relation between the two annotation types. This relation is created when we build the ontology where "الأذان" (Aladan) inform "أوقات الصلاة" (Prayer Times).

Example 4. Searching using the word "الأذان" (Aladan) as keyword.

Search results for the keyword "الأذان" (Aladan) in the "solat-corpus" corpus. The search interface shows 24 results on page 1. The results table includes columns for Left context, Match, Right context, Features, Query, and Document. A summary table on the right shows the following annotation types and counts:

Annotation Type	Count
Token	42342
Lookup	585
صلاة فرض	192
روائي	171
فرض عين	169
صلاة التطوع	147
صلاة العبد	68
مكونات الصلاة	58
السهو	57

Figure 5.5: Search Using the Word "الأذان" (Aladan) as Traditional Way

In Figure 5.5, when we search using the word "الأذان" (Aladan) as keyword (traditional way) the search process returns little documents comparison with using "الأذان" (Aladan) as annotation type (ontology class) in the process of search. Also the search using the word "الأذان" (Aladan) returns the documents that contain the word it self "الأذان" (Aladan) but search using annotation type "الأذان" (Aladan) returns the documents contain "الأذان" (Aladan) and the synonyms and stemming words for it. Additionally the search process returns the ontology components that have relation with the ontology class "الأذان" (Aladan).

5.4 System Evaluation

System evaluation depends on finding all related documents to the ontology components. We use 100 documents in our related Arabic Ontology Domain "فقّه الصلاة" (Prayer jurisprudence) then we used the Gate tool to automatically annotate these documents, based on the Onto Root Gazetteer annotator.

We depend on two important measures which are commonly used to evaluate such a system: precision and recall [44].

Recall: is defined as the number of relevant documents retrieved by a search divided by the total number of existing relevant documents (which should have been retrieved).

Precision: is defined as the number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search.

$$\text{Recall} = \frac{\text{Number of retrieved relevant documents}}{\text{Number of relevant documents}} \quad \dots \text{eq (5.1)}$$

$$\text{Precision} = \frac{\text{Number of retrieved relevant documents}}{\text{Number of retrieved documents}} \quad \dots \text{eq (5.2)}$$

Table 5.1 shows the calculated values of Precision and Recall for the ontology concepts ("رواتب" (Roateb), "صلاة التطوع" (Voluntary Prayer), "فرض عين" (FardAin), "صلاة العيد" (Eid Prayer), "صلاة أهل الأعداء" (Prayer of Exempted People), "مكونات الصلاة" (Prayer Components), "الأذان" (Aladan), "السهو" (Omission), "صلاة فرض" (Obligatory Prayer), "أوقات الصلاة" (Prayer Times), "شروط الصلاة" (Prayer Conditions)). The results are calculated based on equations 5.1 and 5.2.

Table 5.1: Precision and Recall Results

Annotation Types	Recall	Precision
رواتب	97.72	100
صلاة التطوع	93.75	95.74
فرض عين	97.82	100
صلاة العيد	85	94.44
صلاة أهل الاعذار	93.75	100
مكونات الصلاة	95.23	71.42
الآذان	95	82.60
السهو	95	95
شروط الصلاة	84.61	73.33
صلاة فرض	98.07	100
أوقات الصلاة	84.61	73.33

Figure 5.6 depicts the recall and precision for every annotation type in Table 5.1.

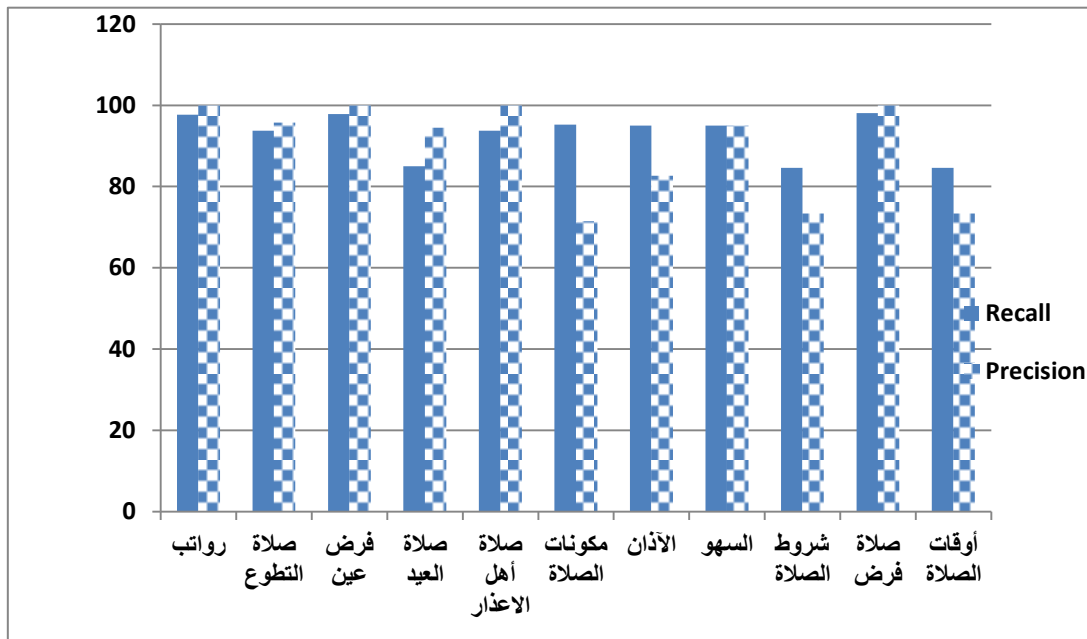


Figure 5.6: Recall and Precision for Every Annotation Type

For comparisons with other methods, since no previous work in Arabic domain using Gate in the process of documents annotation and retrieval and other methods found. So we cannot compare our methodology with other researches.

5.5 Discussion

The results shown in Table 5.1, show high values and low values for different annotation types. This is due to the following reasons:

- For all experiments when we select the corpus of 100 documents we classify these documents where every group of documents related with annotation type (ontology class) to help us in our work for example 1-8 documents contain ontology components that come under annotation type "صلاة فرض" (Obligatory Prayer). We now know when we search using annotation type "صلاة فرض" (Obligatory Prayer) the specified documents will be retrieved because it is known previously and the other documents related with annotation type "صلاة فرض" (Obligatory Prayer) will be retrieved also. This gives some high values and also others low values at the same time.

Another example is that 94-100 documents contain ontology components that come under annotation type "الأذان" (Aladan). So when we search using annotation type "الأذان" (Aladan) the specified documents will be retrieved. Also the documents under annotation type "أوقات الصلاة" (Prayer Times) will be retrieved because there is a relation between the two annotation types "الأذان" (Aladan) and "أوقات الصلاة" (Prayer Times) and this gives high values and low values in our results at the same time.

Also the following can be noted:

- The experiments on a corpus of 100 documents achieved the highest accuracy for every annotation type in our model.
- Precision and Recall may be give different values depending on the size of a corpus of documents.
- Extending and enriching the ontology with more components which are used in the process of document annotation give more comprehensive retrieved documents and accurate results.
- From all our experiments, we can say our system model achieved the best results for all annotation types as we indicated and shown in Table 5.1 and Figure 5.4.
- We can confirm that our system model is better than the traditional ways in the process of documents search and retrieved by giving the best results.

5.6 Summary

In this chapter we have talked about realization, experimental results and evaluation of the proposed system. In the first section we presented the tools and programs used in our work. In the second section we explained the system interface. In the third section we presented the experimental examples performed for some annotation types. The fourth section presented the evaluation measurements for our model. In the fifth section we discussed the results.

Chapter 6: Conclusion and Future Work

We have developed a model for Automatic ontology based document annotation for Arabic information retrieval that facilitates information retrieval with high precision and high recall. This ontology-based model uses ontology components for matching user requests and documents rather than keyword to keyword matching.

Our model consists of several stages: preparing a corpus, building Arabic ontology domain "فقه الصلاة" (Prayer jurisprudence), documents annotation, processing annotated documents, indexing and search process.

Experiments were performed depending on the annotation types presented in Table 5.1 which considers the output of our system automatic ontology-based document annotation for Arabic information retrieval.

For evaluation purposes, the two common effective measures were used Recall and Precision. The Results of annotation types give high Precision and high Recall for all the annotation types as we said in Table 5.1.

Using our system model we overcome the problem of the traditional way used in the process of documents search and retrieval. This means saving time and returns better results.

Our contribution in this work includes the following:

- Adaptation of GATE to work with Arabic documents specially when we use lucene Datastore search engine.
- Building and evaluating a domain specific ontology "فقه الصلاة" (Prayer jurisprudence)
- Building automatic ontology-based document annotation for Arabic information retrieval model used in the process of documents annotation and retrieval.
- A model that covers an important issue in the field of "فقه الصلاة" (Prayer jurisprudence) for users who are interested in the part of Islamic issues.

This work can be improved in multiple directions:

- Extending the ontology by adding the other parts that have relation with the domain "فقہ الصلاة" (Prayer jurisprudence) to include other issues related with Islam.
- Increasing our corpus of documents to retrieve more documents in the domain and obtain more accurate results.
- Extending our system model to be online, have direct relation with the internet this will help to retrieve more and new documents in the field we work in it. This requests building in independent system out of the Gate environment.

References:

1. Vossen G., Lytras MD., and Koudas N., "Editorial: Revisiting the (Machine) Semantic Web: The Missing Layers for the Human Semantic Web". IEEE Trans. Knowl. Data Eng., 19(2): 145-148.
2. Beseiso M., Ahmad A., and Ismail R., "An Arabic Language Framework for Semantic Web". 2011 International Conference on Semantic Technology and Information Retrieval, pp.7 – 11.
3. W3C Resource Description Framework, www.w3.org/TR/REC-rdf-syntax/ (2001). <last accessed August 23, 2012>
4. OWL Web Ontology Language Overview, www.w3.org/TR/owl-features/. (2004)..<last accessed August 23, 2012>
5. Annotations: <http://www.ontotext.com/kim/semanticannotation.html>.retrieved December 06,2007.
6. Giannopoulos G., Bikakis N., Dalamagas T., and Sellis T. "GoNTogle: A Tool for Semantic Annotation and Search". In Proc. of the ESWC 2010 (Demo).
7. Al-Khalifa H., and Al-Wabil, A. (2007). "The Arabic Language and the Semantic Web: Challenges and Opportunities". International Symposium on Computers and the Arabic Language. November 2007, Riyadh, Saudi Arabia.
8. Noy N., and Guinness D., "Ontology Development 101: A Guide to Creating Your First Ontology Stanford University, Stanford, CA, 94305".
9. Welcome to protégé. <http://protege.stanford.edu/overview/protege-owl.html>. <last accessed August 26, 2012>.
10. Welcome to Lucene. [Online] <http://lucene.apache.org> .<last accessed August 26, 2012>.
11. General Architecture for Text Engineering (Gate): <http://gate.ac.uk/>, retrieved December 06, 2007).<last accessed August 26, 2012>
12. Rujiang B., and Xiaoyue W.,” A Semantic Information Retrieval System Based on KIM” , 2010 International Conference on E-Health Networking, Digital Ecosystems and Technologies,2010, pp. 392-395.
13. QI H., ZHANG L., and GAO Y., “Semantic Retrieval System Based on Corn Ontology” , 2010 Fifth International Conference on Frontier of Computer Science and Technology,2010.
14. Zaidi S., and Laskri M.”A cross-language information retrieval based on an Arabic ontology in the legal domain”. The International Conference On Signal-Image Technology & Internet–Based Systems (SITIS’05), Morocco, 2005.

15. Zhai j., and Zhou K., "Semantic Retrieval for Sports Information Based on Ontology and SPARQL", 2010 International Conference of Information Science and Management Engineering, pp. 395 – 398.
16. Matousek K., and Kouba Z., "ODCA – Ontology-Based Document and Content Annotation in Structural Health Monitoring" , Database and Expert Systems Applications (DEXA), 2011 22nd International Workshop on,2011,pp. 302 – 305.
17. Huang M., Wang D., and Wang K.," Ontology-based semantic retrieval of XBRL data", 2011 International Conference on Business Computing and Global Informatization, pp. 363 – 366.
18. Gao H., Zhao J., Yin Q., and Wang J.," Ontology-based Enterprise Information Retrieval Model",2009 International conference on Grey Systems and Intelligent Services, pp. 1326 – 1330.
19. Aufaure M., Soussi R., and Baazaoui H.,"SIRO: On-Line Semantic Information Retrieval using Ontologies",pp. 321 - 326 .
20. Kiryakov A., Popov B., Ognyanoff D., Manov D, Kirilov A., and Goranov M.,"Semantic annotation, indexing, and retrieval". Elsevier's Journal of Web Semantics, 2, 2005.
21. Cunningham H., Maynard D., Bontcheva K., and Tablan V., "Gate: and Architecture for Development of Robust HLT Application". 40th Anniver- sary Meeting of the Association for omputational Linguistics- ACL'02, 2002.
22. Popov B., Kiryakov A., Kirilov A., Manov D., Ognyanoff D., and Goranov M. "Kim -semantic annotation platform". Ontotext Lab, Sirma AI EOOD, 138 Tsarigradsko Shose, Sofia 1784, Bulgaria.
23. Vargas-Vera M., Motta E., Domingue J., Lanzoni M., Stutt A., and Ciravegna F., "Mnm:Ontology driven semi-automatic and automatic support for semantic markup". The 13th International Conference on Knowledge Engineering and anagement (EKAW2002), pages 379-391, 2002. Spain.
24. Domingue J., Dzbor M., and Enrico M., "Magpie: Browsing and navigating on the semantic web". Proceedings ACM Conference on Intelligent User Interfaces (IUI), pages 191-197, January 2004. Portugal.
25. Siegfried Handschuh, Ste@en Staab, and Fabio Ciravegna. "S-cream: Semi-automatic creation of metadata. Semantic Authoring, Annotation & Knowledge Markup". Preliminary Workshop Programme, 2002.
26. Jiang J., Wang Z., Liu C., Tan Z., Chen X., and Li M.,"The Technology of Intelligent Information Retrivela Based on the Semantic Web" Signal Processing Systems (ICSPS), 2010 2nd International Conference on ,pages 824-827. China.
27. Mustafa J., Khan S., and Latif K., "Ontology Based Semantic Information Retrieval". 2008 4th International IEEE Conference "Intelligent Systems".

28. Kassim J., and Rahmany M., "Introduction to Semantic Search Engine".2009 International Conference on Electrical Engineering and Informatics 5-7 August 2009, Selangor , Malaysia.
29. Ditze H., and Schroeder M., "Go Web: A semantic search engine for the life, science Web.2009.
30. Ding L., Finin T., Joshi A. et al "Swoogle: A Semantic Web Search and Metadata Engine", Proc. of the Thirteenth ACM Conference on Information and Knowledge Management, 2004.
31. Pan J., and Horrocks I., "RDFS(FA): Connecting RDF(S) and OWL DL", Knowledge and Data Engineering, IEEE Transactions on, 2007, pages 192-206.
32. He G., and An L., "Ontology Language OWL Research Study", Management and Service Science (MASS), 2011 International Conference on, pages 1- 4.
33. Hu B., Wang J., and Zhou Y., " Ontology Design for Online News Analysis", Intelligent Systems, 2009. GCIS '09. WRI Global Congress on, pages 202-206.
34. Agarwal P., "Semantic Web In Comparison to Web2.0 ", 2012 Third International Conference on Intelligent Systems Modelling and Simulation, pages 558-563.
35. Fang W., Zhang L., Wang Y., and Dong S., "Toward A semantic Search Engine Based on Ontologies", Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005, pages 1913-1918.
36. Tang S., and Cai Z., "Tourism Domain Ontology Construction from the Unstructured Text Documents", pages 297-301.
37. موسوعة الفتاوي , موقع اسلام ويب . <http://www.islamweb.net/ver2/Fatwa/index.php>.
38. Pellicer F., Blazquez L., Iso J., Corcho O., Bernabe M., and Rodriguez A." Using a hybrid approach for the development of an ontology in the hydrographical domain ".
39. Wikipedia, <http://en.wikipedia.org/wiki/Stemming> . ,<last accessed: April 19,2013>
40. Gate, <http://gate.ac.uk/sale/tao/splitch13.html#x18-34000013.8> , <last accessed April 19, 2013>.
41. Wikipedia, <http://en.wikipedia.org/wiki/Tokenization> ,<last accessed April 19, 2013>.
42. Gate, <http://gate.ac.uk/sale/tao/splitch8.html#chap:jape> ,<last accessed April 19,2013>.
43. Wikipedia,http://en.wikipedia.org/wiki/Search_engine_indexing ,<last accessed April 19,3013>.
44. M. S. Al Tayyar "Arabic information retrieval system based on morphological analysis (AIRSMA)" Ph.D. Thesis DeMonfort University July 2000.
45. Brank J., Grobelnki M., and Maldenic D., "A Survey of Ontology Evaluation Techniques", Jozef Stefan Institute, Jamova 39, Slovenia.

46. Christopher B., Harith A., Srinandan D., and Orick W., "Data Driven Ontology Evaluation". In International Conference on Language Resources and Evaluation, Lisbon, Portugal, 24 - 30 May 2004.
47. Antoniou G., and Hormelen F., "A Semantic Web primer", The MIT Press Cambridge, Massachusetts London, England.
48. Kim J., and Choi S., "Evaluation of Ontology Development Methodology with CMM-i", Fifth International Conference on Software Engineering Research, Management and Applications, 2007, pages 823 – 82.
49. Subhashini R., "A survey on Ontology Construction Methodologies", International Journal of Enterprise Computing and Business Systems, January 2011.
50. Hakia, <http://www.hakia.com> ,<last accessed June 11, 2013>
51. Wikipedia, <http://en.wikipedia.org/wiki/Synonym>, <last accessed June 11 ,2013>
52. Hiemstra, D. Using language models for information retrieval. PH.D. Thesis. The Netherlands.(2000).
53. Gate, <http://gate.ac.uk/sale/tao/splitch8.html#x12-2470008.9>,<last accessed July 23, 2013>.
54. Wartena C., Brussee R., Gazendam L., " Apolda: A Practical Tool for Semantic Annotation", Database and Expert Systems Applications, 2007. DEXA '07. 18th International Workshop on , pages 288 – 292.

Appendix A: Part of OWL Source Code

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>
<rdf:RDF xmlns="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#"
  xml:base="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <owl:Ontology rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2"/>
  <!--
  //////////////////////////////////////
  //
  // Object Properties
  //
  //////////////////////////////////////
  -->
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2# تخفف_فيها -->
  <owl:ObjectProperty rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2# تخفف_فيها">
    <rdfs:label rdf:datatype="&xsd:string">تسهل</rdfs:label>
    <rdfs:label rdf:datatype="&xsd:string">تقل</rdfs:label>
    <rdfs:label rdf:datatype="&xsd:string">تيسر</rdfs:label>
    <rdfs:range rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#شروط_صحة"/>
    <rdfs:domain rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#صلاة_أهل_الاعذار"/>
  </owl:ObjectProperty>
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2# لا_ترتبط -->
  <owl:ObjectProperty rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2# لا_ترتبط">
    <rdfs:label rdf:datatype="&xsd:string">لا تتضمن</rdfs:label>
    <rdfs:label rdf:datatype="&xsd:string">لا تشمل</rdfs:label>
```

```

<rdfs:range rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الأذان"/>
<rdfs:domain rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#صلاة_العبد"/>
</owl:ObjectProperty>
<!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#مرتبطه_بصلاة -->
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#مرتبطه_بصلاة">
  <rdfs:label rdf:datatype="&xsd:string">ترتبط بصلاة</rdfs:label>
  <rdfs:label rdf:datatype="&xsd:string">تكون قبل</rdfs:label>
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#رواتب"/>
  <rdfs:range rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#فرض_عين"/>
</owl:ObjectProperty>
<!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#يؤذن -->
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#يؤذن">
  <rdfs:label rdf:datatype="&xsd:string">يخير
</rdfs:label>
  <rdfs:label rdf:datatype="&xsd:string">يعلم
</rdfs:label>
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الأذان"/>
  <rdfs:range rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#وقت_الصلاة"/>
</owl:ObjectProperty>
<!--
////////////////////////////////////
//
// Classes
//
////////////////////////////////////
-->
<!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#أركان -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#أركان">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#مكونات_الصلاة"/>
</owl:Class>
<!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الأذان -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الأذان">
  <rdfs:label rdf:datatype="&xsd:string">المؤذن</rdfs:label>
  <rdfs:label rdf:datatype="&xsd:string">مؤذن</rdfs:label>
  <rdfs:label rdf:datatype="&xsd:string">يؤذن</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#فقه_الصلاة"/>

```

```

</owl:Class>
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#السهو -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#السهو">
  <rdfs:label rdf:datatype="&xsd:string">النسيان</rdfs:label>
</rdfs:label>
  <rdfs:label rdf:datatype="&xsd:string">سهو</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الفقه_الصلاة_2"/>
</owl:Class>
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#رواتب -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#رواتب">
  <rdfs:label rdf:datatype="&xsd:string">سنن الراتبة</rdfs:label>
</rdfs:label>
  <rdfs:label rdf:datatype="&xsd:string">سنن الرواتب</rdfs:label>
</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#صلاة_التطوع_2"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#مرتبطة_بصلاة_2"/>
      <owl:allValuesFrom rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#فرض_عين_2"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#رواتب_بعديه_2 -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#رواتب_بعديه_2">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#رواتب_2"/>
</owl:Class>
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#رواتب_قبليه_2 -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#رواتب_قبليه_2">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#رواتب_2"/>
</owl:Class>
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#سهو_زيادة_2 -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#سهو_زيادة_2">
  <rdfs:label rdf:datatype="&xsd:string">سهو الزيادة</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#السهو_2"/>
</owl:Class>

```

```

<!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#سهو_شك -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#سهو_شك">
  <rdfs:label rdf:datatype="&xsd:string">سهو_شك</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#السهو"/>
</owl:Class>

<!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#سهو_نقصان -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#سهو_نقصان">
  <rdfs:label rdf:datatype="&xsd:string">سهو_نقصان
</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#السهو"/>
</owl:Class>

<!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#شروط_الصلاة -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#شروط_الصلاة">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#فقه_الصلاة"/>
</owl:Class>

<!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#شروط_صحة -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#شروط_صحة">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#شروط_الصلاة"/>
</owl:Class>

<!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#شروط_وجوب -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#شروط_وجوب">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#شروط_الصلاة"/>
</owl:Class>

<!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#صلاة_أهل_الاعذار -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#صلاة_أهل_الاعذار">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#فقه_الصلاة"/>
</owl:Class>

<!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#صلاة_التطوع -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#صلاة_التطوع">
  <rdfs:label rdf:datatype="&xsd:string">تطوع</rdfs:label>
  <rdfs:label rdf:datatype="&xsd:string">صلاة_تطوع</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#فقه_الصلاة"/>
</owl:Class>

<!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#صلاة_العید -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#صلاة_العید">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#صلاة_التطوع"/>

```

```

</owl:Class>
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#صلاة_فرض -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#صلاة_فرض">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#فقه_الصلاة"/>
</owl:Class>
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#فرض_عين -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#فرض_عين">
  <rdfs:label rdf:datatype="&xsd:string">المفروضة</rdfs:label>
  <rdfs:label rdf:datatype="&xsd:string">المكتوبة</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#صلاة_فرض"/>
</owl:Class>
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#فرض_كفاية -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#فرض_كفاية">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#صلاة_فرض"/>
</owl:Class>
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#فقه_الصلاة -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#فقه_الصلاة">
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#مبطلات -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#مبطلات">
  <rdfs:label rdf:datatype="&xsd:string">بطلان</rdfs:label>
  <rdfs:label rdf:datatype="&xsd:string">يبطل</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#مكونات_الصلاة"/>
</owl:Class>
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#مستحبات -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#مستحبات">
  <rdfs:label rdf:datatype="&xsd:string">مستحب</rdfs:label>
  <rdfs:label rdf:datatype="&xsd:string">يستحب
</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#مكونات_الصلاة"/>
</owl:Class>
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#مكروهات -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#مكروهات">
  <rdfs:label rdf:datatype="&xsd:string">الكرهه</rdfs:label>
  <rdfs:label rdf:datatype="&xsd:string">كراهة_الصلاة</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#مكونات_الصلاة"/>
</owl:Class>

```

```

<!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الصلاة_مكونات -->

<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الصلاة_مكونات">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الصفة_الصلاة"/>
</owl:Class>

<!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#وقت_الصلاة -->
<owl:Class rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#وقت_الصلاة">
  <rdfs:label rdf:datatype="&xsd:string">أوقات</rdfs:label>
  <rdfs:label rdf:datatype="&xsd:string">أوقات الصلاة</rdfs:label>
</rdfs:label>
  <rdfs:label rdf:datatype="&xsd:string">مواقيت الصلاة</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الصفة_الصلاة"/>
</owl:Class>

<!--
////////////////////////////////////
//
// Individuals
//
////////////////////////////////////
-->

<!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#استقبال_القبلة -->
<owl:NamedIndividual rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#استقبال_القبلة">
  <rdfs:type rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#شروط_صحة"/>
</owl:NamedIndividual>

<!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الإستخارة -->
<owl:NamedIndividual rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الإستخارة">
  <rdfs:type rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#صلاة_التطوع"/>
</owl:NamedIndividual>

<!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الإستسقاء -->
<owl:NamedIndividual rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الإستسقاء">
  <rdfs:type rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#صلاة_التطوع"/>
  <rdfs:label rdf:datatype="&xsd:string">استسقاء</rdfs:label>
  <rdfs:label rdf:datatype="&xsd:string">استسقى</rdfs:label>
  <rdfs:label rdf:datatype="&xsd:string">فاسقتنا</rdfs:label>
  <rdfs:label rdf:datatype="&xsd:string">يستسقى</rdfs:label>

```

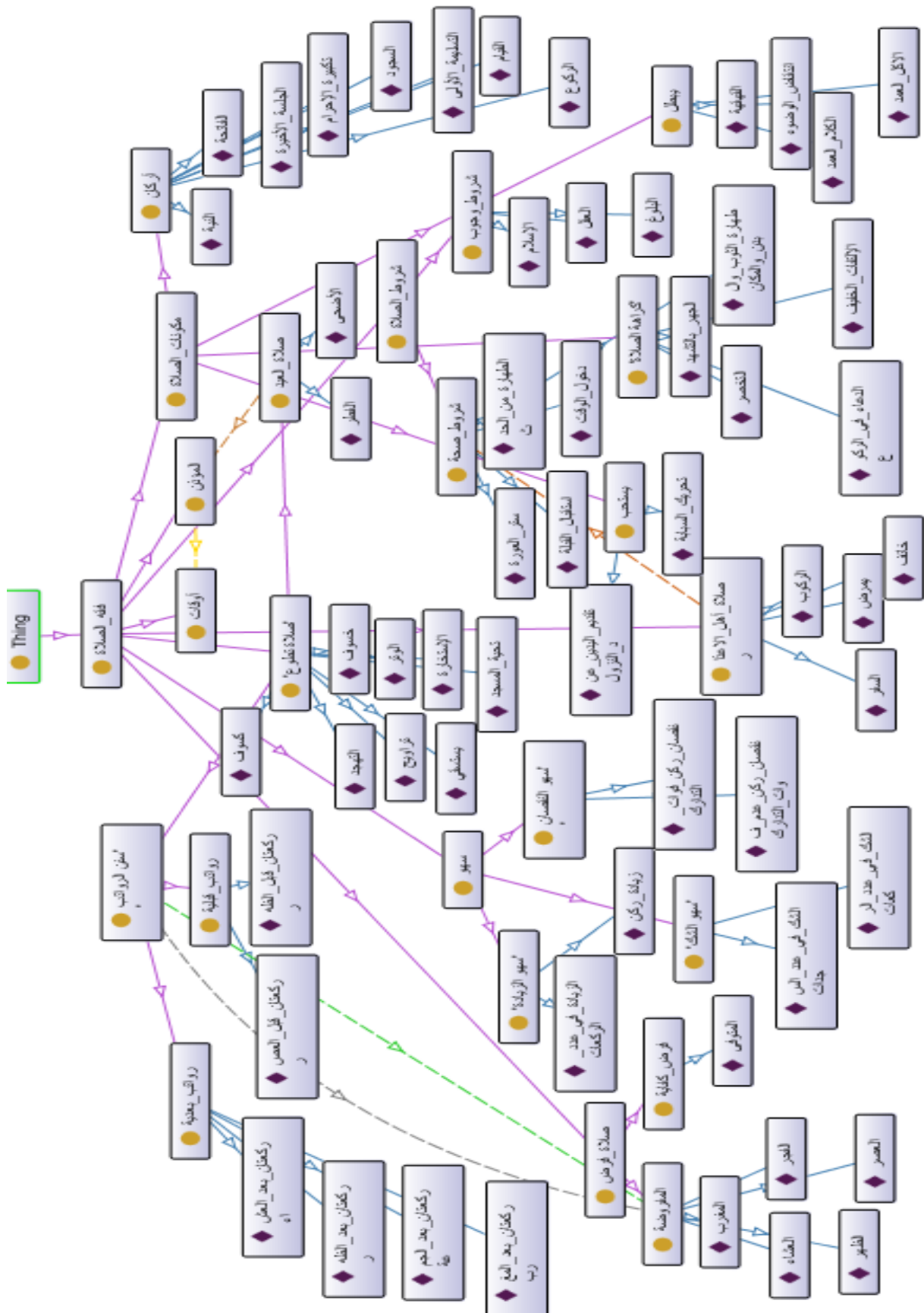
```

</owl:NamedIndividual>
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الإسلام -->
<owl:NamedIndividual rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الإسلام">
  <rdf:type rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#شروط_وجوب"/>
</owl:NamedIndividual>
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الإلتفات_الخفيف -->
<owl:NamedIndividual rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الإلتفات_الخفيف">
  <rdf:type rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#مكروهات"/>
</owl:NamedIndividual>
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الاضحى -->
<owl:NamedIndividual rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الاضحى">
  <rdf:type rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#صلاة_العبد"/>
  <rdfs:label rdf:datatype="&xsd:string">الأضحى</rdfs:label>
</owl:NamedIndividual>
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الاكل_العمد -->
<owl:NamedIndividual rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الاكل_العمد">
  <rdf:type rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#مبطلات"/>
</owl:NamedIndividual>
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#البلوغ -->
<owl:NamedIndividual rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#البلوغ">
  <rdf:type rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#شروط_وجوب"/>
</owl:NamedIndividual>
  <!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#التخصر -->
<owl:NamedIndividual rdf:about="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#التخصر">
  <rdf:type rdf:resource="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#مكروهات"/>
</owl:NamedIndividual>

<!-- http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#التراويح -->

```


Appendix B: Ontology Graph



Appendix C: Example of Jape Rules

Phase: salataheladr

Input: Token Lookup

Options: control = appelt

Rule: salataheladr1

```
(
{Lookup.URI=="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#صلاة_أهل_الأعدار-2"}
):label
-->
{
gate.AnnotationSet label = (gate.AnnotationSet)bindings.get("label");
gate.Annotation personAnn = (gate.Annotation)label.iterator().next();
gate.FeatureMap features = Factory.newFeatureMap();
features.put("rule", "salataheladr1");
outputAS.add(label.firstNode(), label.lastNode(), "صلاة أهل الأعدار",
features);
}
```

Rule: salataheladr1

```
(
{Lookup.URI=="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الخانف"}
):label
-->
{
gate.AnnotationSet label = (gate.AnnotationSet)bindings.get("label");
gate.Annotation personAnn = (gate.Annotation)label.iterator().next();
gate.FeatureMap features = Factory.newFeatureMap();
features.put("rule", "salataheladr1");
outputAS.add(label.firstNode(), label.lastNode(), "صلاة أهل الأعدار",
features);
}
```

Rule: salataheladr1

```
(
{Lookup.URI=="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الراكب"}
):label
```

```

-->
{
gate.AnnotationSet label = (gate.AnnotationSet)bindings.get("label");
gate.Annotation personAnn = (gate.Annotation)label.iterator().next();
gate.FeatureMap features = Factory.newFeatureMap();
features.put("rule", "salataheladr1");
outputAS.add(label.firstChild(), label.lastNode(), "صلاة أهل الأعدار",
features);
}
Rule: salataheladr1
(
{Lookup.URI=="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#المريض"}
):label
-->
{
gate.AnnotationSet label = (gate.AnnotationSet)bindings.get("label");
gate.Annotation personAnn = (gate.Annotation)label.iterator().next();
gate.FeatureMap features = Factory.newFeatureMap();
features.put("rule", "salataheladr1");
outputAS.add(label.firstChild(), label.lastNode(), "صلاة أهل الأعدار",
features);
}
Rule: salataheladr1
(
{Lookup.URI=="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#المسافر"}
):label
-->
{
gate.AnnotationSet label = (gate.AnnotationSet)bindings.get("label");
gate.Annotation personAnn = (gate.Annotation)label.iterator().next();
gate.FeatureMap features = Factory.newFeatureMap();
features.put("rule", "salataheladr1");
outputAS.add(label.firstChild(), label.lastNode(), "صلاة أهل الأعدار",

```

```

features);
}
Rule: salataheladr1
(
{Lookup.URI=="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#تخفف_فيها"}
):label
-->
{
gate.AnnotationSet label = (gate.AnnotationSet)bindings.get("label");
gate.Annotation personAnn = (gate.Annotation)label.iterator().next();
gate.FeatureMap features = Factory.newFeatureMap();
features.put("rule", "salataheladr1");
outputAS.add(label.firstChild(), label.lastNode(), "صلاة أهل الأعدار",
features);
}
Rule: salataheladr1
(
{Lookup.URI=="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#شروط_صححة"}
):label
-->
{
gate.AnnotationSet label = (gate.AnnotationSet)bindings.get("label");
gate.Annotation personAnn = (gate.Annotation)label.iterator().next();
gate.FeatureMap features = Factory.newFeatureMap();
features.put("rule", "salataheladr1");
outputAS.add(label.firstChild(), label.lastNode(), "صلاة أهل الأعدار",
features);
}
Rule: salataheladr1
(
{Lookup.URI=="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#استقبال_القبلة"}
):label

```

```

-->
{
gate.AnnotationSet label = (gate.AnnotationSet)bindings.get("label");
gate.Annotation personAnn = (gate.Annotation)label.iterator().next();
gate.FeatureMap features = Factory.newFeatureMap();
features.put("rule", "salataheladr1");
outputAS.add(label.firstChild(), label.lastNode(), "صلاة أهل الأعدار",
features);
}
Rule: salataheladr1
(
{Lookup.URI=="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#الطهارة_من_الحدث-2"}
)
:label
-->
{
gate.AnnotationSet label = (gate.AnnotationSet)bindings.get("label");
gate.Annotation personAnn = (gate.Annotation)label.iterator().next();
gate.FeatureMap features = Factory.newFeatureMap();
features.put("rule", "salataheladr1");
outputAS.add(label.firstChild(), label.lastNode(), "صلاة أهل الأعدار",
features);
}
Rule: salataheladr1
(
{Lookup.URI=="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#دخول_الوقت-2"}
)
:label
-->
{
gate.AnnotationSet label = (gate.AnnotationSet)bindings.get("label");
gate.Annotation personAnn = (gate.Annotation)label.iterator().next();
gate.FeatureMap features = Factory.newFeatureMap();
features.put("rule", "salataheladr1");

```

```

outputAS.add(label.firstChild(), label.lastNode(), "صلاة أهل الأعدار",
features);
}
Rule: salataheladr1
(
{Lookup.URI=="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#ستر_العورة-2"}
)
:label
-->
{
gate.AnnotationSet label = (gate.AnnotationSet)bindings.get("label");
gate.Annotation personAnn = (gate.Annotation)label.iterator().next();
gate.FeatureMap features = Factory.newFeatureMap();
features.put("rule", "salataheladr1");
outputAS.add(label.firstChild(), label.lastNode(), "صلاة أهل الأعدار",
features);}
Rule: salataheladr1
(
{Lookup.URI=="http://www.semanticweb.org/ashraf/ontologies/2012/8/untitled-ontology-2#طهارة_الثوب_والبدن_والمكان-2"}
)
:label
-->
{ gate.AnnotationSet label = (gate.AnnotationSet)bindings.get("label");
gate.Annotation personAnn = (gate.Annotation)label.iterator().next();
gate.FeatureMap features = Factory.newFeatureMap();
features.put("rule", "salataheladr1");
outputAS.add(label.firstChild(), label.lastNode(), "صلاة أهل الأعدار",
features);}

```