

**The Islamic University–Gaza
Research and Postgraduate Affairs
Faculty of Information Technology
Master of Information Technology**



**الجامعة الإسلامية - غزة
شؤون البحث العلمي والدراسات العليا
كلية تكنولوجيا المعلومات
ماجستير تكنولوجيا المعلومات**

Extraction of Taxonomic Relations from Arabic Text for Ontology Construction

**استخلاص العلاقات التصنيفية من النص العربي لغرض بناء
الأنطولوجيا**

Basel Salama El Zraie

Supervised by

Dr. Rebhi S. Baraka

Associate prof. of Computer Science

**A thesis submitted in partial fulfilment
of the requirements for the degree of
Master of Information Technology**

July/2016

إقرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

Extraction of Taxonomic Relations from Arabic Text for Ontology Construction

استخلاص العلاقات التصنيفية من النص العربي لغرض بناء الأنطولوجيا

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

Declaration

I understand the nature of plagiarism, and I am aware of the University's policy on this.

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

Student's name:	باسل سلامة الزريعي	اسم الطالب:
Signature:		التوقيع:
Date:	28/6/2016	التاريخ:

Abstract

The huge amount of textual information available electronically has made it difficult for many users to search and find the right information within acceptable time. The ontology based techniques can contribute to solve these problems and help users in exploiting these vast resources. Ontology could be an efficient way to improve the process of searching and exploiting information on the web. The benefit of ontology is that it provides a standard for the vocabulary used in a specific domain and relations. This thesis proposes a method to extract taxonomic relations to construct ontology automatically from natural Arabic text on Political News domain using four stages. First perform pre-processing operations in text such as tokenization, normalization and stop-word removing and then morphological information in pre-processing is extracted to detect the part of speech of each word. Second extraction of terms by integration between lexical resources and machine-learning classifier for Arabic named entities recognition. Third extraction of taxonomic relations between terms using rule based domain. Finally constructing a set of transformation rules to identify the appropriate ontological elements from the terms and taxonomic relations that extracted. After constructing the ontology, we build RDF language to represent information about resources on the text and build ontology with class-subclass relations and property relations. Two methods are performed to test and evaluate the accuracy of approach, first using measures calculate precision, recall and f-measure. Second using a reasoner to check the consistency. The results shows satisfactory results for all terms and taxonomic relations extraction, with precision = 92% and recall = 91%.

Keywords Automatic Ontology Construction, Arabic NLP, Taxonomic Relation Extraction, Named Entities Recognition.

المخلص

كمية المعلومات النصية الهائلة الموجودة إلكترونياً جعلت عملية البحث وإيجاد المعلومة المطلوبة في الوقت المناسب عملية صعبة. ساهمت تقنية الانطولوجيا في حل هذه المشكلة ومساعدة المستخدمين في استغلال الموارد الهائلة حيث يمكن ان تكون وسيلة فعالة لتحسين عملية البحث واستغلال المعلومات علي شبكة الإنترنت. الأنطولوجيا تزودنا بمعايير للمفردات المستخدمة في مجالات محددة والعلاقات بين هذه المفردات. هذه الرسالة تقترح طريقة لاستخراج العلاقات التصنيفية لغرض بناء الانطولوجيا آلياً من النص العربي في مجال الاخبار السياسية، وذلك باستخدام أربعة مراحل. أولاً اجراء عمليات ما قبل المعالجة للنص العربي مثل تقسيم النص الي كلمات، تسوية وتوحيد اشكال الحروف، ازالة واستبعاد بعض الكلمات، واستخراج مصدر كل كلمة واقسام الكلمات. وفي المرحلة الثانية نستخرج المصطلحات الموجودة في النص عن طريق تكامل بين المصادر المعجمية والتعليم الالي للتعرف علي الاسماء. ثالثاً استخراج العلاقات التصنيفية بين المفردات باستخدام قواعد في مجال محدد. واخيراً بناء قواعد التحويل لاستخراج عناصر الانطولوجيا المناسبة من المصطلحات والعلاقات التصنيفية المستخرجة. وبعد ذلك نبنى لغة إطار وصف المصادر "Resource Description Framework" لتمثيل المعلومات الموجودة في النص، وبناء الانطولوجيا علي شكل اصناف وعلاقات. تم بإجراء اختبارين لتقييم طريقة بناء الانطولوجيا، أولاً من خلال حساب قياسات دقة النتائج. ثانياً التحقق من الاتساق باستخدام المحقق "Reasoner". حيث ان النتائج حققت نتائج مرضية في مجال استخراج المصطلحات والعلاقات التصنيفية، حيث اعطت نسبة دقة وهي 92% ونسبة إرجاع 91%.

الكلمات الأساسية: بناء الانطولوجيا آلياً، معالجة اللغة العربية، استخراج العلاقات التصنيفية، تحديد الأسماء من النص.

Dedication

To my parents ...

To my sisters and brothers ...

To my teachers ...

To my friends ...

To Palestine ...

Acknowledgment

First and foremost, thanks to Allah for giving me the power and help to accomplish this research. Without the grace of Allah, I was not able to accomplish this work.

*Many thanks and sincere gratefulness goes to my supervisor **Dr. Rebhi S. Baraka**, for his help, guidance, and continuous follow-up in this research.*

Special thanks also to my parents, my sisters and brothers for their endless support. Without them, I would never have been able to achieve my goals.

Table of Contents

Declaration	I
Abstract.....	II
الملخص.....	III
Dedication	IV
Acknowledgment.....	V
Table of Contents	VI
List of Tables	X
List of Figures.....	XI
List of Abbreviations	XIII
Chapter 1 Introduction.....	2
1.1 Statement of the Problem	4
1.2 Objectives.....	4
1.2.1Main Objective	4
1.2.2Specific Objectives	4
1.3 Importance of the Thesis	5
1.4 Scope and Limitations.....	5
1.5 Research Methodology	6
1.6 Thesis Organization	7
Chapter 2 Theoretical and Technical Foundation.....	9
2.1 Ontology.....	9
2.2 Structure and Components of Ontology.....	10
2.3 Ontology learning	12
2.4 Ontology Representation.....	13
2.4.1 Resource Description Framework.....	14
2.4.2 Web Ontology Language	15
2.5 General Architecture for Text Engineering	16
2.5.1 GATE Component Model.....	16
2.5.2 JAPE Component.....	18
2.5.3 Semantic Annotation.....	18
2.6 Named Entity Recognition.....	19
2.6.1 NER Approaches.....	19
2.7 Formal Definition for Discovering Taxonomic Relations	20

2.8	Performance Evaluation.....	21
2.8.1	Accuracy.....	21
2.8.2	Precision.....	21
2.8.3	Recall.....	21
2.8.4	F-measure.....	21
2.9	Summary.....	22
Chapter 3 Related Works.....		24
3.1	Named Entity Recognition.....	24
3.1.1	Rule Based and Statistical Approach.....	24
3.1.2	Machine Learning-Based Approach.....	26
3.1.3	Hybrid Approach.....	27
3.2	Relations Extraction.....	28
3.3	Ontology Construction and Learning.....	31
3.3.1	Automatic Arabic Ontology Constructing.....	31
3.3.2	Automatic English Ontology Construction.....	33
Chapter 4 Automatically Constructing Domain Ontology from Arabic Text.....		37
4.1	Approach Overview.....	37
4.2	Pre-processing Stage.....	38
4.2.1	Preparing the Corpus.....	39
4.2.2	Encoding.....	39
4.2.3	Tokenization.....	40
4.2.4	Normalization.....	40
4.2.5	Stop-Word Removal.....	41
4.2.6	Sentence Splitting.....	42
4.3	NLP and Features Extraction.....	42
4.3.1	Part-Of-Speech Tagging.....	43
4.3.2	Morphological Analysis.....	44
4.4	Terms Extraction Stage.....	45
4.4.1	Lexical Resources for NER.....	46
4.4.2	Transducer.....	47
4.4.3	Machine Learning Based NER.....	47
4.5	Taxonomic Relation Extraction.....	49
4.5.1	Defining the Semantic Taxonomic Relation Category.....	50
4.5.2	Discovering the Actual Patterns.....	51

4.5.3 Searching for Instances of s Relation using Patterns	52
4.6 Transforming to Ontological Elements and Knowledge Representation.....	52
4.7 Summary.....	52
Chapter 5 Implementation.....	54
5.1 Tools and Programs	55
5.2 Pre-processing	55
5.2.1 Datasets.....	55
5.2.2 Encoding.....	56
5.2.3 Normalization.....	56
5.2.4 Stop-Word Removal	57
5.2.5 Sentence Splitting	57
5.2.6 Tokenization.....	58
5.2.7 POS Tagging.....	59
5.2.8 Light Stemming	60
5.3 Terms Extraction.....	61
5.3.1 Lexical Resources	61
5.3.2 Machine Learning Based NER	63
5.4 Taxonomic Relations Extraction.....	64
5.5 Transformation of Annotated Text into Ontological Elements	68
5.6 Summary.....	71
Chapter 6 Experimental Results and Evaluation.....	73
6.1 Experimental Setup	73
6.2 Arabic News documents Corpus.....	73
6.3 Data Pre-processing Results.....	74
6.4 Terms Extraction Result.....	75
6.5 Taxonomic Relations Extraction Result	76
6.6 Ontology visualizer and Language Presentation.....	77
6.7 Evaluation of the Approach	79
6.7.1 Domain Expert Review VS the Proposed Approach	79
6.7.2 Named Entities Recognition and Human Evaluation	79
6.7.3 Taxonomic Relations Extraction and Human Evaluation.....	81
6.7.4 Using Reasoner	85
6.8 Summary.....	87
Chapter 7 Conclusions and Future Work.....	89

7.1 Summary.....	89
7.2 Contribution	90
7.3 Future Work	90
References.....	91
Appendix: JAPE Rules for Ontology Construction.....	96

List of Tables

Table (4.1): Stop-Word List Sample.....	41
Table (4.2): Common Syntactic Categories.....	43
Table (4.3): Categories of Named Entity.....	46
Table (4.4): Category Lists in Gazetteer.....	46
Table (4.5): Trigger Word.....	47
Table (4.6): Taxonomic Relationships.....	51
Table (5.1): BBC Arabic Corpus Details.....	56
Table (5.2): Rules of Taxonomic Relations.....	65
Table (5.3): Transformation of annotated text into ontological elements	65
Table (6.1): Summary of Evaluation Based on the Domain Expert and the Proposed Approach for Extracting Named Entities.....	80
Table (6.2): Summary of Evaluation Based on the Domain Expert and the Proposed Approach for Extracting Taxonomic Relations	83
Table (6.3): Summary the Results of Calculation R, P and F-measure for Extracting Taxonomic Relations	84

List of Figures

Figure (2.1): Ontology of Plants	10
Figure (2.2): Ontology Learning from Text Layer Cake	13
Figure (2.3): Web Ontology Language	14
Figure (2.4): Resource Description Framework Triple	15
Figure (2.5): Semantic Annotation	19
Figure (4.1): The Approach to Construct Ontology from Text	38
Figure (4.2): Pre-processing Stage	39
Figure (4.3): Tokenization Process	40
Figure (4.4): Features Extraction Stage	42
Figure (4.5): Terms Extraction Stage	45
Figure (5.1): Sentence Splitting in Gate	58
Figure (5.2): Tokenization Process in Gate	59
Figure (5.3): Part-of-Speech Features in Gate	60
Figure (5.4): Stemming Features in Gate	61
Figure (5.5): Gazetteer Resource in GATE	62
Figure (5.6): JAPE Rule for Taxonomic Relation Creation " is-a"	66
Figure (5.7): JAPE Rule for Building Triple Statements	67
Figure (5.8): JAPE Rule to Create Ontological Concepts and Resources	69
Figure (5.9): Classes and Subclasses for Political News Ontology	70
Figure (5.10): Ontological Properties for Taxonomic Relations	71
Figure (6.1): Annotation Set from System and Domain Expert	74
Figure (6.2): Set of Processing Resources for Pre-processing Stage	75
Figure (6.3): Name Entity Extraction	76
Figure (6.4): Sample of Name Entity Annotation	76
Figure (6.5): Sample of Taxonomic Relations Extraction	77
Figure (6.6): Classes and subclasses in news ontology	78
Figure (6.7): RDF triples as based on the ontology	78
Figure (6.8): Document from BBC News to Named Entity Recognition Evaluation	80
Figure (6.9): Named Entity Recognition Evaluation Using Annotation Diff	81
Figure (6.10): Document from BBC News to Taxonomic Relations Evaluation	82
Figure (6.11): Taxonomic relations evaluation using Annotation Diff	83

Figure (6.12): Taxonomic Relations Evaluation Using Corpus Quality Assurance in GATE.....	84
Figure (6.13): Consistency Ontology	85
Figure (6.14): Consistency for the Properties of Taxonomic Relations	86
Figure (6.15): OWLViz Displaying the Asserted Hierarchy for the Ontology	86

List of Abbreviations

ATRC	Annual Text Retrieval Conferences
CRF	Conditional Random Fields
DAG	Directed Acyclic Graph
GATE	General Architecture for Text Engineering
GOLD	General Ontology for Linguistic Description
IE	Information Extraction
IR	Information Retrieval
JAPE	Java Annotation Patterns Engine
LHS	Left Hand Side
LRs	Language Resources
ME	Maximum Entropy.
ML	Machine Learning
NER	Named Entity Recognition
NEs	Name Entities
NIST	National Institute of Standards and Technology.
NLP	Natural Language Processing
OL	Ontology Learning
OSAC	Open Source Arabic Corpora
OWL	Web Ontology Language
OWL DL	Web Ontology Language- Description logic
PCFGs	Probabilistic Context-Free Grammars
POS	Part of Speech
PRs	Processing Resources
RDF	Resource Description Framework
RHS	Right Hand Side
SL	Supervised Learning
SSL	Semi-Supervised Learning
SVM	Support Vector Machines
SW	Semantic Web
TF-IDF	Term Frequency Inverse Document Frequency.
URI	Uniform Resource Identifiers
VRs	Visual Resources

Chapter 1

Introduction

Chapter 1

Introduction

Arabic language is of essential importance to Muslims because it is the language of the Quran and the mother tongue of 23 countries. However, Arabic content on the Web although limited and less than other languages, it is increasing rapidly and is represented as information and web pages based knowledge in Arabic documents (Albukhitan & Helmy, 2013). Users are facing the problem of finding relevant information in the Arabic content. One of the major reasons is that most search engines find matches based on keywords without consideration of their meanings. To overcome this issue in search engines and information retrieval, semantic web technologies play an important role for meaningful retrieval of information on the web. The semantic web, is widely expected to facilitate semantic matching between the user query and the indexed documents based on ontology.

Ontologies are suggested as a knowledge representation that is capable of expressing sets of entities, relationships, properties and axioms of a given domain. Manually constructed ontologies often have some challenges, they are difficult and time consuming process (Ribeiro de Azevedo et al., 2014). Many efforts have been exerted for constructing ontologies and to overcome the bottleneck of knowledge extraction, but the majority of these methods have focused mainly on English or Latin languages like found in (Al Arfaj & Al Salman, 2014) (De Azevedo et al.; Wang, Li, Bontcheva, Cunningham, & Wang, 2006) (Wang et al., 2006) (Zayaraz, 2015) (Correia, Girardi, & de Faria, 2011). Other languages such as Arabic language still need more research to improve this field. So, there are need for building Arabic ontology with automatic approaches that are considered more suitable for building large scale ontologies where challenges of time and efforts of human experts become a bottleneck. According to Gruber, Ontologies are formal and explicit specifications of shared conceptualizations in the form of concepts and relations. The ontology is used as a conceptual infrastructure to represent a given domain as concepts and relationships between these concepts, and can thus be seen as an explicit specification of a conceptualization (Gruber, 1993). Ontologies are basically

semantic containers and capable to describe the set of terms, relationship between terms and axioms in a given domain or corpus.

Generally, terms or entities are extracted by patterns of simple and complex nouns or machine learning to named entity recognition in large text corpora. Relations can be extracted by simple verbs between entities or lexical patterns. Algorithms for such tasks can be used dynamically by various machine learning approaches on large text corpora (Pandit, 2010). Extracting relationships and entities enables us to build ontology from text and representing it by Web Ontology Language (OWL) (Bechhofer, 2009) and Resource Description Framework (RDF) (Manola, 2004).

Taxonomic relations is a collection of controlled vocabulary terms organized into a hierarchical structure. Each term in a taxonomy is one or more parent-child relationships to other terms in the taxonomy. Taxonomies are useful relations for organizing many aspects of knowledge. As components of ontologies, the main paradigms of taxonomy learning are on the one hand pattern based approaches and on the other hand distributional hypothesis based approaches (Ryu & Choi, 2006). The former are approaches based on matching lexico-syntactic patterns which convey taxonomic relations in a corpus (Hearst, 1992), and the latter are statistical approaches based on the distribution of context in the corpus.

The approach for automatic ontology construction is relies on the extraction of domain concepts (terminology) and categorization of these concepts by processing natural language text and predefined list for NEs. After that concept is linked with lexico patterns as taxonomic relations, then transform concepts to classes and subclasses relations and taxonomic relations to property relations as ontological elements. Therefore both information extraction and text mining are important for ontology construction. The news reports about Political News in the Middle East are extracted and annotate accordingly by specifying Arabic location, organization and person positions by a taxonomic relations. For example, given the Arabic statement: " فلسطين عضو في جامعة الدول العربية ". First we extract the ontological terms: "فلسطين" and "جامعة الدول العربية" where these terms are extracted using machine learning and

training based on classified data lists that contain locations and organizations. Next the taxonomic relation "عضو في" is extracted using lexical patterns.

Next, we state the problem of the research, the objectives we aim to achieve, the importance of our thesis, scope and limitations and then the research methodology we follow to achieve the research objectives and hence solve the research problem.

1.1 Statement of the Problem

Constructing ontology is very important part of semantic applications and manually constructing ontologies is a difficult and a time consuming process and often involves domain experts. This process is more difficult with Arabic language which has various morphological and syntactic variations.

The problem of this research is how to construct ontologies with taxonomic relations from Arabic text on a Political News domain using automatic method.

1.2 Objectives

We organize the objectives into the main objective which reflects the research problem and the specific objectives which presents the functional phases that if achieved would lead to solving the research problem.

1.2.1 Main Objective

To build a approach for automatically constructing domain ontology from Arabic text by the extraction of terms that exist in text documents and the identification of the taxonomic relationships that hold between them, with achieving the required level of accuracy.

1.2.2 Specific Objectives

1. To collect appropriate set of documents on specific domain and perform the required pre-processing such that these documents can be used as a basis to extract features such as terms, characteristics and relationships.
2. To collect and use linguistic resources as predefined lists of Named Entities (NEs) to be used in machine learning as supervised learning, to extract terms.
3. Named entity recognition, that is capable of recognizing different instances of NEs types: Person, Location, Organization, Nationality etc.

4. Acquisition regular expression to detect hyponyms, has-a, is-a, part-whole, kind-of automatically by constructing lexical patterns of knowledge between concepts in text in order to facilitate information extraction from Arabic text.
5. To represent extracted concepts and relations in appropriate ontology representations such as RDF and RDFS.
6. To conduct performance evaluation on the proposed approach for accuracy.

1.3 Importance of the Thesis

- Arabic ontologies to be constructed can be used in information retrieval in specific domains as an attempt to improve both recall and precision of the search.
- This work can be helpful in identify, extract and represent relationships in order to facilitate comprehensive information extraction from unstructured text.
- Identifying the Named Entities such as Person, Location and Organization Names etc. from the text can be used as a pre-processing step for several Natural Language Processing systems.
- The research contributes in the newly starting area of automatic ontology construction in Arabic domains. It is likely to encourage research in this area based on the above values.

1.4 Scope and Limitations

- Unstructured text such as plain text documents is considered for extracting terms and stating relations between them.
- Specific domain is chosen which is Political News as the domain of the ontology to be constructed.
- In constructing the ontology, we limit relations between extracted terms to direct taxonomic relations at the level of RDFS such as Is-A, Cause-Effect, Part-Whole, Has-A, Kind-Of relations and excluding complex relations at the level of OWL such as symmetric/asymmetric, disjointness, cardinality relations.
- Ignore words in Latin characters as non-Arabic words in processing the Arabic text, because NER is deal with Arabic words in extracting named entity.

- The ontology language used are RDF, RDFS and OWL without any restrictions.

1.5 Research Methodology

The approach we follow in this thesis to achieve our objective as follow:

Phase 1: Data collection and pre-processing using GATE tool.

- Collect collection of documents text about specific domain in Political News as coups, that contain 2350 documents about BBC Arabic news. We will divided the dataset into three parts, where the first part is used in training for machine learning and the second part is used to develop the model, third part used to test the approach.
- Sentence splitting by punctuation marks like comma ",", period ".", using GATE.
- Tokenization: The process to split text into words that called tokens. The list of tokens becomes input for further processing such as parsing.
- Normalize some character by standard character as Substituting letters.

Phase 2: Features Extraction.

- Identifying part-of-speech tagging using grammatical parser such as Arabic Stanford parser.
- Perform morphological analysis (Light Stemming) by deletion of prefixes and suffixes character to identify the root word.

Phase 3: Terms Extraction.

- Populate Gazetteer by predefined list type to named entites recognition.
- Fulfilling machine learning to enhanced named entity recognition. Use this technique to generate a classification model for classified token within texts into predefined types as class, such as Person, Location and Organization names. The feature set is selected to develop the ML-based component is (current word, previous word, next word, POS tags, word length, stemmer).

Phase 4: Taxonomic relations extraction.

Taxonomic relations extraction involves applying an appropriate rules based pattern-matching. Patterns are discovered by querying the underlying text using JAPE rule that produces a sequence of words that involves taxonomic relations between terms. The taxonomic relations category is: (Is-A, Cause-Effect, Part-Whole, Has-A, Kind-Of).

Phase 5: Ontology Building.

It involves constructing a set of Transformation JAPE rules, which are used to identify appropriate ontological element from the texts. Ontological classes and subclass relations can be automatically extracted using the named entity recognitions and their categorizations. Ontological properties relations that bind terms automatically extracted using taxonomic relations extraction.

Phase 6: Evaluation the approach using two methods. First human expert in a Political News domain to define the suitable terms and relations. Takes three directions: measuring the correctness of extracted patterns with respect to existing correct ones using a recall metric, measuring the ability of our proposed methodology to detect patterns with respect to all retrieved information using a precision metric, and, finally, applying an f-measure that denotes the overall accuracy. Second using a reasoner to check the ontology consistency.

1.6 Thesis Organization

The research is organized as follows: Chapter 1 is the Introduction. Chapter 2 is Theoretical and Technical Foundation. Chapter 3 Related Works. Chapter 4 presents the approach for Automatically Constructing Domain Ontology from Arabic text. Chapter 5 is about the Implementation. Chapter 6 presents Experimental Results and Evaluation. Chapter 7 is dedicated for the Conclusions and Future Work.

Chapter 2

Theoretical and Technical Foundation

Chapter 2

Theoretical and Technical Foundation

In this chapter, the fundamental concepts which represent the basis for understanding our research are presented. First, Ontology definition and components is defined, and then shows how ontology is constructed and learning from input text as (structured, semi-structured or unstructured data) providing ontology language to represent the ontology. After that, we provide an overview to Named Entity Recognition (NER) that is used in terms extraction and the development environment used in this thesis. Then we provide formal definitions used in extracting taxonomic relations. Finally, we present an overview of the used performance evaluation approach.

2.1 Ontology

Ontologies are basically semantic containers and capable to describe the set of terms, relationship between terms and axioms in a given domain or corpus. Ontologies specify the vocabulary of all possible terms used in the specific domain and the relationships that may exist between these terms. It is generally defined by Gruber in (Gruber, 1993) as "Ontology is a formal, explicit specification of a shared conceptualization in the form of concepts and relations". Explicit word to denote terms and the relationships between them, a conceptualization word can be described as an abstract representation of the world or domain we want to model for a certain purpose.

Another definition can be found in (Blomqvist, 2005), as "A hierarchically structured set of concepts describing a specific domain of knowledge that can be used to create a knowledge base. Ontology contains concepts, a subsumption hierarchy, arbitrary relations between concepts, and axioms. It may also contain other constraints and functions". Figure (2.1) depicts an example ontology of plants that consists of set of classes and subclasses, where the superclass is "Plants" that consists of two subclasses ("Vegetable", "Fruit") and there subclasses is superclass for others subclasses such as ("Citrus", "Berry", "Stem"). There are properties in

ontology plants for classes such as "Type" and also individuals such as ("Lemon", "Potato").

Formal definitions of ontology is presented by (Bozsak et al., 2002) as "An ontology is a structure $O := (C, \leq_C, R, \leq_R)$ consisting of:

- Two disjoint set C and R called concept identifiers and relation identifiers respectively.
- partial order \leq_C on C called concept hierarchy or taxonomy.
- Function $\sigma: R \rightarrow C \times C$ called signature.
- Partial order \leq_R on R called relation hierarchy.

To be used by any system, the ontology must be formally defined in term of its information structure and format of its representation.

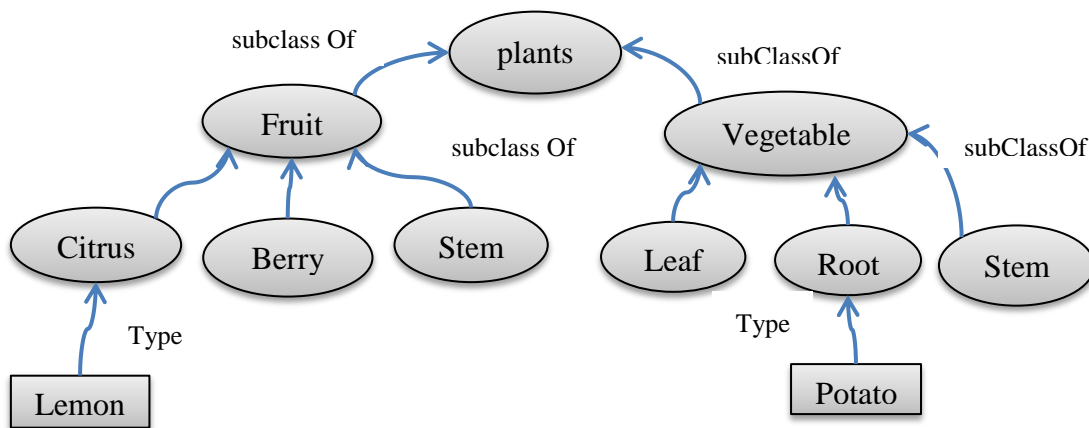


Figure (2.1): Ontology of Plants

2.2 Structure and Components of Ontology

Ontology consists of individuals, classes, attributes, and relations. Individuals are the basic components of an ontology. The individuals in an ontology may include concrete objects such as people, animals and plants, as well as abstract individuals such as numbers and words. Classes are the sets or collections of objects describe by the set of attributes. Classes may classify individuals with help of these attributes. Some examples of classes are Person, Vehicle, Car, Thing, etc. Attributes are properties and features that classes can have. For example, a person class or object has the properties name, age, height, etc. Relationships between objects in an ontology specify how objects are related to other objects. For example in the

ontology that contains the concept "Motor-Vehicle" and the concept "Vehicle" might be related by a relation of type "is a"(Ahmed, 2009).

2.3 Relations type in ontology

Relations define the interactions between entities or concept and typically classified as taxonomic or non-taxonomic relations.

1. Taxonomic relations:

Taxonomic involve putting each concept in the correct place in a hierarchy, it usually a simple hierarchical arrangement of entities. This considered to be an important task in the ontology learning process, since it provides the taxonomic layer of the ontology including equivalence, hypynomy, parent/child, subClass/superClass or broader/narrower forms (Nakashole, Weikum, & Suchanek, 2012). As ball example, ball could be said to be a hyponym of sports equipment (is a) in a sports domain, while blue is a color and the concept ball has a color.

2. Non taxonomic relations:

Non taxonomic relations is arbitrary complex relations between concepts and expected to have a single verb connecting two entities such as *A worksFor B*. This can representing how one concept can act upon another in the given domain. For the ball example, a player can kick a ball. The relations learning involves finding relationships among concepts. (Cimiano & Völker, 2005).

Common approaches to extracting taxonomic relations are lexico-syntactic patterns, agglomerative hierarchical clustering, distributional similarity and formal concept analysis (FCA).

- Lexico-syntactic patterns define a pattern on lexical annotations on a corpus which are likely to represent instances of particular relations. An example of such a pattern for English is NP such as NP, NP and NP. Lexico-syntactic pattern tend to give high precision but low recall because of the variety of ways these relations can be expressed in natural language.
- Agglomerative Hierarchical Clustering of concepts builds a hierarchy of clusters, starting with each concept as a distinct cluster. Each clustering step compares each pair of clusters according to some similarity measure, and the

pair with the highest similarity are merged. This repeats until some predicate is satisfied.

- Distributional Similarity in its simplest form asserts that there exists a relationship between concepts which occur within some bounded context. The strength of the relationship depends on the frequency of their co-occurrence. For example, if concept A only occurs in the presence of concept B, and concept B occurs more frequently than concept A, we might infer that A and B are related and that B is more general than A.
- Formal Concept Analysis FCA considers the attributes which apply to each concept. By analyzing the attributes concepts share, a lattice of commonality and subsumption can be construct.

2.4 Ontology learning

Ontology Learning (OL) is an automated or semi-automated process to construction of ontologies from domain data in which ontological elements such as concepts and relations are extracted automatically from different resources (Shamsfard & Barforoush, 2003). Another definition of OL refers to extracting conceptual knowledge from structured, semi-structured or unstructured data. Structured data, such as databases, have semantics described by its schema or structure. Semi-structured data such as wikis. Unstructured data are in the form of plain text and depend on pre-processing techniques from the field of Natural Language Processing to provide syntactic annotations like part of speech or syntactic dependencies. OL methods are then applied to the annotated corpus, each method extracting one or more kind of ontology element.

Buitelaar and Cimiano (Buitelaar et al., 2005) suggests an ontology learning layer cake as shown in Figure (2.2). This ontology learning layer cake can be used to classify an OL approach according to the task that it aims at. These tasks are described below:

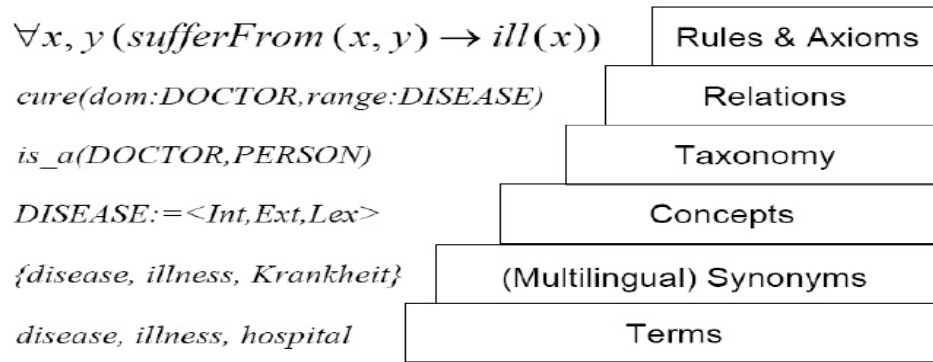


Figure (2.2): Ontology Learning from Text Layer Cake (Buitelaar, Cimiano, & Magnini, 2005)

- **Term extraction** extracts the relevant phrases and terms for a specific domain. Typically, a textual documents or corpus is used as the input for term extraction.
- **Synonym discovery** used to find synonym words for concepts and acquisition of semantic term variants between languages. This definition is similar to the synsets in WordNet, for this task WordNet is used to discover and extract synonym.
- **Concept formation** defines concept to provide an intentional definition of the concept, set of concept instances and set of linguistic realizations.
- **Concept hierarchies** involve putting each concept in the correct place in a hierarchy. This is considered to be an important task in the ontology learning process since it provides the taxonomic layer of the ontology.
- **Relations learning** involves finding relationships among concepts. There are different types of relations, for example, in the case of binary relations appropriate domain and range have to be identified.
- **Rules** are concerned with the axiomatic definition of concepts. The task in this layer is to learn the rules that apply for concepts and relations. For example, learn which pairs of concepts are disjoint

The OL tasks are ordered in the way that each layer is built depending on the output of the lower layer, i.e., a concepts hierarchy learning task can only be achieved if the appropriate concepts are first extracted.

2.5 Ontology Representation

According to the definitions of ontology, ontology is used in describing a domain of knowledge. Consequently, this domain of knowledge needs to be represented in a

machine understandable language in-order to perform basic operations such as query or storage. This ontology languages can formally describe the meaning of terminology used in web documents. Ontology languages are created at the beginning of the 1990's. Figure (2.3) summarizes the hierarchy of different ontology languages (Corcho et al., 2003).

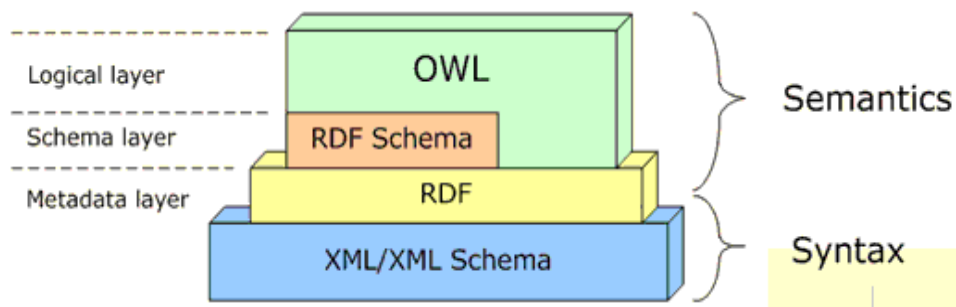


Figure (2.3): Web Ontology Language (Corcho, Fernández-López, & Gómez-Pérez, 2003)

Various languages are developed to represent ontology, we describe the most common languages for ontology representation.

2.5.1 Resource Description Framework

Resource Description Framework (RDF) is a language used for representing information about resources on the web. It is a basic ontology language. RDF is written in XML. By using XML, RDF information can easily be exchanged between different types of computers using different types of operating systems and application languages. RDF was designed to provide a common way to describe information so it is machine readable. RDF descriptions are not designed to be displayed on the Web (Champin, 2001). Data model for objects and relations between them, provides a simple semantics for data model. Data models can be represented in XML syntax. RDF identifies resources with Uniform Resource Identifiers (URI) (Corcho et al., 2003). The base element of the RDF model is the triple: a subject linked through a predicate to object. In RDF triple (S,P,O) We say that <subject>has a property <predicate>valued by <object>, as example " فلسطين عضو " العربية الجامعة العربية", for the triple " فلسطين " is subject, " عضو في " is predicate, " الجامعة العربية " is object.

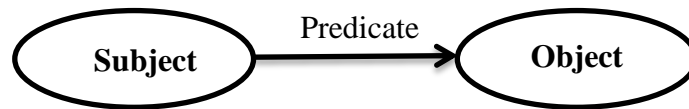


Figure (2.4): Resource Description Framework Triple

2.5.2 Web Ontology Language

Web Ontology language (OWL) is created in 2001 by a Web-Ontology (WebOnt) Working Group. The aim of this group was to make a new ontology mark-up language for the Semantic Web (McGuinness & Van Harmelen, 2004). OWL is used when the information contained in documents needs to be processed by application. OWL can be used to explicitly to represent the meaning of terms in vocabularies and the relationships between the terms. OWL adds more vocabulary for describing properties and classes. In this thesis we used owl to represent classes and subclasses that extracted from text and properties of relations between classes.

Siblings of OWL are OWL Lite, OWL DL and OWL Full.

2.5.2.1 OWL Lite

OWL Lite supports classification hierarchy and simple constraints. OWL Lite provides a quick migration path for thesauri and other taxonomies. OWL Lite has a lower formal complexity than OWL DL.

2.5.2.2 OWL DL

Maximum expressiveness while retaining computational completeness and decidable i.e. all computations will be finished in time. OWL DL is named due to its correspondence with Description Logic, and it includes all the OWL language constructs.

2.5.2.3 OWL Full

OWL Full gives syntactic freedom of RDF, with no computational guarantees. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. OWL Full can be viewed as an extension to RDF. whereas OWL Lite and OWL DL can be viewed as an extension of a restricted view of RDF. Every OWL (Lite, DL, Full) document is an RDF document and every RDF

document is an OWL Full document. Only some RDF documents can be OWL Lite or OWL DL.

2.6 General Architecture for Text Engineering

In this section, we present tool used in nature language processing and there functionality and components.

2.6.1 GATE Component Model

General Architecture for Text Engineering (GATE) (Maynard et al., 2001) is one of the most popular freely available software tools dealing with NLP. GATE is a suite of Java tools that provides an infrastructure for developing and deploying software components that process human language. The motivating factors behind choosing the GATE is include reusability of components, task-based evaluation, robustness, efficiency, and portability; the tools support Arabic languages; GATE components consists of three types Language Resources (LRs) to represent lexicons such as corpora and ontologies, Processing Resources (PRs) to provides a set of essential tools for NLP system development including tokenizers, gazetteers, POS taggers, chunkers, parsers, an OrthoMatcher component, and a grammar, all of which are used within a simple Arabic rule-based NER application built as a part of GATE. It facilitates the development of rule-based NER systems by providing the user with the capability of implementing grammatical rules as a finite state transducer using JAPE (Java Annotation Patterns Engine). Visual Resources (VRs) represent visualization components (Maynard et al., 2001). 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. GATE consists of tools for NLP system development:

- **Tokenizers**

The tokenizer component splits the text into very simple tokens such as numbers, punctuation and words of different types, each split is called token. The following kinds of Token are possible:

- **Word:** Is defined as any set of contiguous upper or lowercase letters, including a hyphen (but no other forms of punctuation).

- Number: Is defined as any combination of consecutive digits. There are no subdivisions of numbers.
 - Symbol: Two types of symbol are defined, currency symbol (e.g. '\$', '£') and symbol (e.g. '&', '^'). These are represented by any number of consecutive currency or other symbols (respectively).
 - Punctuation: Three types of punctuation are defined: start_punctuation (e.g. '('), end_punctuation (e.g. ')'), and other punctuation (e.g. ':'). Each punctuation symbol is a separate token.
 - Space Token: White spaces are divided into two types of Space Token space and control according to whether they are pure space characters or control characters.
- **Gazetteer**

A gazetteer consists of a set of predefined lists containing names of entities such as cities, organisations, person name, etc. These lists are used to find occurrences of these names in text, e.g. for the task of named entity recognition. The word "gazetteer" is often used interchangeably for both the set of entity lists and for the processing resource that makes use of those lists to find occurrences of the names in text. When a gazetteer processing resource is run on a document, annotations of type Lookup are created for each matching string in the text.

- **Sentence Splitter**

The sentence splitter is a cascade of finite-state transducers which segments the text into sentences. The splitter uses a gazetteer list of abbreviations to help distinguish sentence-marking full stops from other kinds. Each sentence is annotated with the type "Sentence". Each sentence break (such as a full stop) is also given a "Split" annotation.

- **POS Tagger**

The POS tagger produces a part-of-speech tag as an annotation on each word or symbol, such as (V) for verbs. The tagger uses a default lexicon and rule set for English language. Arabic language require external lexicon such as Arabic Stanford tagger.

- **OrthoMatcher**

The OrthoMatcher module adds identity relations between named entities found by the semantic tagger, in order to perform coreference resolution within the document.

2.6.2 JAPE Component

Java Annotation Patterns Engine (JAPE) is part of GATE. It is specially developed pattern matching language for GATE over annotations based on regular expressions. JAPE makes it possible to recognise complex regular expressions in annotations on documents. A JAPE grammar consists of a set of phases, each of which consists of a set of pattern/action rules. The phases run sequentially and constitute a cascade of finite state transducers over annotations. The left hand side (LHS) of the rule contains the identified annotation pattern that may contain regular expression operators (e.g. *, ?, +). The right hand side (RHS) outlines the action to be taken on the detected pattern and consists of annotation manipulation statements (Thakker, Osman, & Lakin, 2009). There is an example to extract team names from text, based on the name of the city followed by certain suffixes:

```
Rule: team_rule
Priority:50
( {City}
  ( {Token.string=="United" } | {Token.string=="F.C." } | {Token.string=="FC" }
  ) ):team
-->
:team.Team = {rule= " team_rule" }
```

2.6.3 Semantic Annotation

Semantic annotation is the process of identifying knowledge elements in text and mapping them to instances and entities in a given knowledge base in GATE. It is the process of automatic generation of named entity annotations with class and instance references to a semantic repository (Maynard et al., 2001). Figure (2.5) shows the semantic annotation process to matching between knowledge element in text and

there entities in a given semantic repository, such as word London match to City in repository.

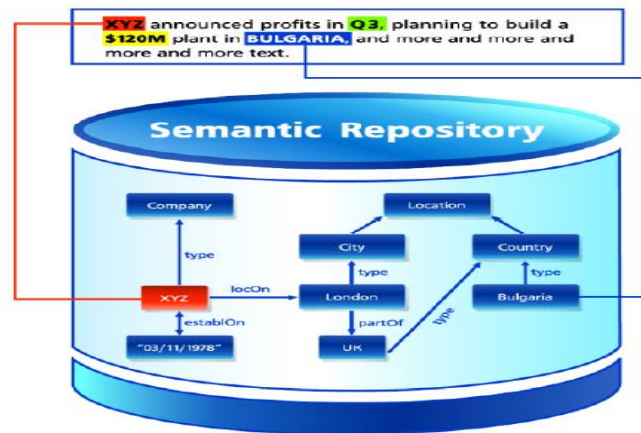


Figure (2.5): Semantic Annotation (Kiryakov et al., 2003)

2.7 Named Entity Recognition

Named Entity Recognition (NER) is considered to be the most fundamental task of any information extraction (IE) system. NER is a task to detecting the Named Entities (NEs) in a document and then categorize these NEs into predefined list of Named Entity classes such as Name of Person, Location, Organization etc. (Nadeau & Sekine, 2007). The main task of NER was broken down into three subtasks: first task Name Entities (NE) - ENAMEX tag to identify proper names (locations, persons, organizations, etc.), second task Temporal Expression - TIMEX tag to identify dates and times, and third task Number Expression - NUMEX tag to identify number and percentages and money in documents (Chinchor & Robinson, 1997).

2.7.1 NER Approaches

Approaches for NER from text, fall under three categories (Shalan, 2014). The first approach known as "rule based NER" combines grammar, in the form of manual rules, with gazetteers to extract named entities. The second, is "machine learning based NER" which utilizes large datasets and features extracted from these, to train a classifier to recognize a named entity. The third approach is "hybrid NER" which combines both of the rule based NER and machine learning based NER.

2.7.1.1 Rule-Based Approach

Rule-based NER systems depend on local handcrafted linguistic rules to extract NEs within texts using linguistic and grammar rules, usually this rules extracted from experts and then encoded as a set of rules (Shaalán, 2010). Such systems using gazetteers/dictionaries to build rule. The rules are usually implemented in the form of regular expressions and heuristic rules to identify names.

2.7.1.2 Machine Learning-Based NER

Machine learning widely used in order to extract NE tagging decisions from annotated texts that are used to generate statistical models for NE prediction. This method depends on classification rules triggered by features with positive and negative examples assigned on previous processed entities. The machine learning approaches used in NER classified to Supervised Learning (SL), the Semi-Supervised Learning (SSL), and the Unsupervised Learning (UL) (Nadeau & Sekine, 2007).

2.7.1.3 Hybrid Approach

The hybrid approach integrates the rule-based approach with the ML-based approach in order to optimize overall performance. The process flow may be from the rule-based approach to the ML-based approach or vice versa (Shaalán, 2014).

2.8 Formal Definition for Discovering Taxonomic Relations

In this thesis, extracting taxonomic relations between entities depends on the following definitions (wikipedia):

Definition 1. If every element in a set A is also a member of set B then A is a subset of B , i.e., $A \subset B$. And if and only if all element in A belongs to the set B and every element in B belongs to set A , i.e., $A \subseteq B$ and $A \supseteq B$.

Definition 2. If A element in a set B then A is belong to B , i.e., $A \in B$.

Definition 3. The universal quantification that symbols " \forall " is a type of quantifier which is interpreted as "for all". It expresses that a propositional function can be satisfied by every member of a domain of discourse.

2.9 Performance Evaluation.

To measure the performance of our method, we use several performance metrics. There are many classification measures like accuracy, precision, recall, and F-measure.

2.9.1 Accuracy

The Accuracy is used to represent the percentage of test set instances that are correctly classified by the classifier.

$$\text{Overall Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (2.1)$$

2.9.2 Precision

Precision is used to represent the percentage of the number of items identified for a given topic as the number of correctly predicted items. The higher the precision, the better the system is correct.

$$\text{Precision} = \frac{TP}{(TP + FP)} \quad (2.2)$$

2.9.3 Recall

Recall is used to represent a percentage of the total number of correct items for a given topic as the number of correctly predicted items.

$$\text{Recall} = \frac{TP}{(TP + FN)} \quad (2.3)$$

2.9.4 F-measure

F-measure is a standard statistical used to measure the performance of system. The F-measure is a conjunction parameter based on precision and recall.

$$F - \text{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.4)$$

2.10 Summary

In this chapter, we presented an overview of the basic theoretical foundation related to our research. We presented overview of ontology, its structure, ontology learning and representation. Since we use the GATE framework for our work, we presented the GATE component model and its structure. Named Entity Recognition (NER) is of essential importance in our research specifically in the process of identifying and extracting terms. The chapter also presented a formal definition for discovering taxonomic relations which helps in stating and using rules. Finally, we stated the most important and relevant performance metrics and classification measures that are used to evaluate the effectiveness of ontology construction.

In the next chapter, we provide various works about automatic ontology construction from Arabic and English texts.

Chapter 3

Related Works

Chapter 3

Related Works

Automatic ontology construction and learning is a knowledge acquisition activity that relies on automatic methods to transform unstructured data sources into conceptual structures.

Automatic ontology construction for the Arabic language, has few efforts where most efforts concentrate on English language. Others adopt a manual or a semi-automatic approach.

In this chapter we review number of research works about automatic ontology construction from Arabic and English text. This literature review is divided into three sections: literature about named entity recognition, relations extraction, and ontology construction and learning.

3.1 Named Entity Recognition

There are set of approaches used in named entity recognition, this approaches are divided into three sections: rule based approach, machine learning based approach, and hybrid approach.

3.1.1 Rule Based and Statistical Approach

Zaidi et. al. (Zaidi, Laskri, & Abdelali, 2010) present a rule base approach to extract structured information in specified domain such as Name Entities by Java Annotation Pattern Engine (JAPE) rules in the Gate framework. Jape rules are used to enhance the Gate tools by extracting terms in the form of collocations from Arabic text such as Noun-Noun, Adjective-Noun, Verb-Noun etc. using Jape rules. The components of Gate are used as language resources (LR) such as documents and corpora, processing resources (PR) such as tokenising and parsing, visual resources (VR) this component for graphical user interface. Jape provides finite state transduction over annotations based on patterns and regular expressions. The system is capable of extracting named entities through predefined patterns that use tokenized and morphology analysed corpus with Part-Of-Speech (POS) features. These features

are used in Jape rules as regular expression in left-hand-side (LHS) of the Jape rule, where LHS consists of an annotation pattern description. Validation is done by a human expert in the domain; he accepts or rejects collocations. AnnotationDiff is used to calculate F-measure, it gives 0.66.

Al-Thubaity et. al. (Al-Thubaity, Khan, Alotaibi, & Alonazi, 2014) present two basic methods to automatically extract single and multi-word terms from Arabic special domain corpora. The methods are based on two simple heuristics. The first method is based on most frequent words, where frequent single words, 2-grams, and 3-grams in special domain corpora are typically terms. The second method is based on terms, either single or compound and they are typically bounded by closed-class words, such as prepositions, determiners, and conjunctions, or by orthographic signs, such as punctuation, numbers, currency, and other symbols. The following steps outline the method. First corpus pre-processing, which includes corpus segmentation, where word prefixes and suffixes are separated using Stanford Arabic Segmenter and then removing Arabic diacritics, numbers, and Latin characters, and normalizing *hmza* and *taa marbutah*. Second, candidate terms identification is done by tokenization of the corpus single-word, 2-gram, and 3-gram lists are generated with their associated frequency in the corpus, this based on the first method. For the second method, single and multi-word terms are selected if they are bounded by closed-class words. Third, ranking the candidate terms by statistical formula (TF-IDF). Finally selecting the top-ranked terms based in the third step. For experiments, the top-ranked 300 single words are selected, 2-grams, and 3-grams based on frequency of occurrence and TF-IDF. For evaluation the author obtained results comparable to previously published studies.

Asharef et. al. (Asharef et al., 2012) develop a rule-based approach (linguistic approach) to Arabic NER system relevant to the crime domain. Based on morphological information, predefined crime and general indicator lists and an Arabic named entity annotation corpus from crime domain, several syntactical rules and patterns of Arabic NER are induced and then formalized. The system involves modules. First pre-processing modules are sentence splitting, tokenization, and POS tagging. Second module is about named entity identification, that involves detection of their boundaries of tokens that belong to a named entity. Final module

classification, using set of grammatical rules and patterns and gazetteer. The result shows that the accuracy of this system is 90%.

However, these researches play an important role in extracting named entities using syntactic, statistical and linguistic rules based approach. They achieve better results in specific domains. The main challenges of them is Arabic language due its highly complex morphology. However, our approach used named entity based rules to enhance the terms extractions.

3.1.2 Machine Learning-Based Approach

Benajiba et al. (Benajiba, Diab, & Rosso, 2008) develop NER system based on Support Vector Machines (SVM). The use set of features in machine learning, this feature are contextual as window of $\pm n$ tokens from the NE of interest, lexical as special markers for tokens that include digits or punctuation, morphological, gazetteers which use three gazetteers for people and locations and organization name, POS tags and BPC, nationality and the corresponding English capitalization. The system was evaluated using ACE Corpora and ANERcorp. They measure the impact of the different features in isolation and combined. The best results were achieved when all the features are considered.

AbdelRahman et al. (AbdelRahman, Elarnaoty, Magdy, & Fahmy, 2010) integrated two ML approaches to handle Arabic NER: namely bootstrapping semi-supervised pattern recognition and Conditional Random Fields (CRF) classifier as a supervised technique, since it is a discriminative probabilistic model, and is used for segmenting and labelling the sequential data. The feature set used with the CRF classifier included word-level features, POS tag, BPC, gazetteers and morphological features. The system was developed to extract 10 types of NEs: Person, Location, Organization, Job, Device, Car, Cell Phone, Currency, Date and Time. The results show that the system outperforms Ling Pipe (Alias-i, 2008) NE recognizer when both are applied to the ANERcorp dataset.

The ML-NER approach had an ability to extract Named entity based on machine learning techniques, and it needs an annotated (tagged) corpus. It is better to choose the machine learning approach if we deal with an unrestricted domain. However, the main drawback of their approach is ambiguity in Arabic texts because

different diacritics represent different meanings. Also lack of resources for Arabic NER where most of the available resources are either very costly or are of low quality. Our proposed work uses machine learning for NER using GATE and external resources for training/testing our classification component.

3.1.3 Hybrid Approach

It is an approach where more than two approaches are used in order to improve the performance of the NER system.

Oudah et al. (Oudah & Shaalan, 2012) develop Arabic NER system using two approaches a rule-based and Machine Learning (ML) based approach. The system consists of two pipelined components: rule-based and ML-based Arabic NER components. The processing consists of three main phases; first rule-based NER phase, second feature engineering phase, i.e. the feature selection and extraction, and third ML-based NER phase. The proposed system is capable of recognizing 11 different types of named entities (NEs): Person, Location, Organization, Date, Time, Price, Measurement, Percent, Phone Number, ISBN and File Name. Author test three ML algorithms; Decision Trees, Support Vector Machines, and Logistic Regression. The features used in ML across all phases are rule-based features, morphological features, POS tag, word length flag, dot flag, capitalization flag, NE type, nominal flag, check classes Gazetteers feature flags. Two types of linguistic resources are collected and acquired: gazetteers (i.e. predefined lists of NEs or keywords) and corpora (i.e. datasets). The performance of the rule-based component is evaluated using GATE built-in evaluation tool, AnnotationDiff.

Bounhas and Slimani (Bounhas & Slimani, 2009) propos a method to extract multi-word terms, where they focus on compound nouns from Arabic specialized corpora. The proposed approach uses linguistic rules based on morphological features and POS tags to parse documents and retrieve candidate terms by extracting compound nouns from Arabic specialized corpora. Statistical measures are used to deal with ambiguities generated by the linguistic tools and to rank candidate terms according to their relevance. The approach is based on the following principles: first combine two types of linguistic approaches, based on morph-syntactic patterns, so to detect compound noun boundaries and use syntactic rules to handle Multi-Word

Terms (MWTs). Second, handle the ambiguities by studying the context of each word. Thus filtering the solutions provided by the morphological analyser by using the tag proposed by the POS tagger. Three tools are used: morphological analyzer, POS tagger and the syntactic parser. They developed a Morph-POS matcher which coordinates tasks of morphological analysis and POS tagging. The results in term of precision are better than other existing approaches.

Hybrid approaches combined hand crafted rule based system and Machine Learning system (our approach falls in this category). The key characteristic of this system is that the processing is done in stages. In the initial phase, the text passes through some hand coded regular expression rules with high probability of being correct. Second depends on machine learning approaches, where integrated is done by feeding the output of the rule-based system as features to machine-learning classifiers. Experimental results confirm that hybrid approach is significantly better than the pure rule-based system or the pure machine-learning classifier. Perhaps it the most similar work to our approach used rule-based systems to provide training labels for machine learning classifier.

3.2 Relations Extraction

Hearst (Hearst, 1992) described a low cost approach for the automatic acquisition of the hyponymy lexical relation from unrestricted text. Where relations are identified as a set of lexico-syntactic patterns that are easily recognizable, that occur frequently and across text genre boundaries. The proposed patterns is in the form: "<Noun> such as <List of Noun phrases>". This method is meant to provide an incremental step toward the larger goals of natural language processing. This approach is complementary to statistically based approaches that find semantic relations between terms. That requires a single specially expressed instance of a relation while the others require a statistically significant number of generally expressed relations. Their recall is very low.

Al Zamil et. al. (Al Zamil & Al-Radaideh, 2014) present a methodology that extracts ontological relationships from Arabic text. Mainly, extract semantic features of Arabic text, propose syntactic patterns of relationships among concepts, and propose a formal model of extracting ontological relations. The authors enhance

version of Hearst's algorithm and resolve the ambiguity of homonyms, and focus on relation extraction rather than on terms extraction. The method consists of four functional components. Firstly pre-processing and feature extraction to be used in detecting textual patterns, different features used to satisfy the requirements of building lexical syntactic patterns of Arabic text, such as POS tag feature, stem and word to deal with original text. Secondly, building lexical syntactic patterns of Arabic text by enhancing version of Hearst's algorithm on Arabic text. Thirdly, expansion phase to avoid having redundant patterns that refer to the same concepts. Finally, pattern filtering and aggregation. The results indicate that the proposed technique is a good candidate for extracting ontological relations from Arabic text, but results showed that the performance among different datasets is not systematic. However, the Newspapers dataset experienced the highest performance compared with other datasets. Alternately, the Blogs dataset experienced the lowest performance.

Ponzetto and Strube (Ponzetto & Strube, 2007) describe the automatic creation of a large scale domain independent taxonomy. Wikipedia categories are used as concepts in a semantic network and labelled the relations between these concepts as *is_a* and *not is_a* relations by using methods based on the connectivity of the network and on applying lexico-syntactic patterns to very large corpora. The process used to extract taxonomy is as follows: firstly, clean the network from meta-categories used for encyclopaedia management. Secondly, refinement of links identification. Thirdly, set of processing methods used to label relations between categories as *Is-a* is based on string matching of syntactic components of the category labels. Fourthly, employ methods relying on the structure and connectivity of the categorization network. Fifthly, applying methods of lexico-syntactic based as Hearst *is_a* relation extraction. Finally, inference by multiple inheritance and transited. The semantic relations are extracted from infoboxes, hyperlinks within info boxes and list of categories that articles belong to. The results are evaluated for the quality of the created resource by comparing them with ResearchCyc one of the largest manually annotated ontologies, as well as computing semantic similarity between words in benchmarking datasets.

Nakashole et. al. (Nakashole, Weikum, & Suchanek, 2012) present PATTY: a large resource of relational patterns that are arranged in a semantically meaningful

taxonomy along with entity-pair instances. The PATTY resource is freely available for interactive access and download. The PATTY system is based on efficient algorithms for frequent item-set mining and can process Web-scale corpora. The author define an expressive family of relation patterns, which combines syntactic features, ontological type signatures, and lexical features. The PATTY taxonomy consists of 350,569 pattern synsets. Random-sampling-based evaluation shows a pattern accuracy of 84.7%.

Grycner & Weikum (Grycner & Weikum, 2014) develop HARPY for discovering and organizing paraphrases of relations between entities by computing a high-quality alignment between the relational phrases of the PATTY taxonomy and the verb senses of WordNet. The resulting taxonomy of relational phrases and verb senses. HARPY contains 20,812 synsets organized into a Directed Acyclic Graph (DAG) with 616,792 hypernymy links.

Fader et. al. (Fader, Soderland, & Etzioni, 2011) introduce two simple syntactic and lexical constraints on binary relations expressed by verbs in English sentences. The syntactic constraint requires the relation phrase to match the POS tag pattern, the pattern limits relation phrases to be either a verb, a verb followed immediately by a preposition, nouns, adjectives, or adverbs ending in a preposition. On other side identify the problems of incoherent and uninformative extraction for open information extraction systems and enforce constraints on binary as verb-based relation phrases in English. The authors implements the constraints in the REVERB Open IE system. REVERB's biggest improvement came from the elimination of incoherent extractions.

Lahbib et. al. (Lahbib, Bounhas, Elayeb, Evrard, & Slimani, 2013) present a hybrid approach for Arabic semantic relation extraction which mixes statistical calculus and linguistic knowledge. The approach extracts noun phrases at the first stage and then transforms them into semantic relations. They vocalized texts to reduce ambiguities by statistical method and propose a new distributional approach for similarity calculus. The experiments is performed in different domains. Three areas are considered: drinks, purification and fasting. The correctly relations extracted in the field of purification exceeded 70%.

One of the basic requirements for any ontology construction is to find relations between the entities of the document. As shown in the literature, the rule-based approach based on syntactic and lexical patterns are more practical and effective in specific domain relations extraction in natural language than machine learning based on SVM model. Our approach is restricted to extract taxonomic relations only, where used rule-based systems as patterns for taxonomic relations extractions.

3.3 Ontology Construction and Learning

Several studies have dealt with such topics as the construction of ontology from Arabic or English language. However, little attention has been paid to Arabic Political News ontology learning.

3.3.1 Automatic Arabic Ontology Constructing

Hazman et. al. (Hazman, El-Beltagy, & Rafea, 2009) develop a method for semi-automatically learning a hierarchal Arabic ontology from web documents for agricultural domain, they extract concepts by noun phrases appearing in the headings of a document and the document's hierarchical structure and is-a relations between concepts. The ontology is constructed through the use of two complementary approaches. The first approach utilizes the structure of phrases appearing in HTML headings while the second uses the hierarchical structure of the HTML headings for identifying new concepts and their taxonomical relationships between seed concepts and between each other.

Albukhitan and Helmy (Albukhitan & Helmy, 2013) propose a method for automatic annotation of the Arabic web resources related to food, nutrition and health domains. It uses linguistic patterns to discover relevant relationships between the named entities in the Arabic web resources. The extracted information is then associated to the corresponding concepts and object properties of the developed ontology to produce the RDF metadata for the corresponding web resources. The automatic annotation process consists of seven main tasks: web Source Acquisition, Tokenization, Normalization, Named Entity Recognizer (NER), Fact Extraction, Fact Cleaning & Validation, Ontology Mapping and Knowledge Based Enrichment. Sets of NEs and relationships are manually extracted from collected corpus. Then,

compared the information output using the precision, recall and f-measure metrics to evaluate the performance.

Al-Rajebah and Al-Khalifa (Al-Rajebah & Al-Khalifa, 2014) present a model to extract ontologies from Wikipedia using a linguistic approach. They apply the proposed approach on the Arabic version of Wikipedia. The semantic relations were extracted from infoboxes, hyperlinks within infoboxes and lists of categories that articles belong to. To evaluate their system, they conducted three experiments which are: validity testing of the ontology according to OWL rules and human judgments from experts and the crowd. The system output achieves an average precision of 65%.

Al-Arfaj and Al-Salman (Al Arfaj & Al Salman, 2014) present a framework for ontology construction from Arabic texts based on Hadith (sayings of prophet Mohammed). They discuss the challenges facing ontology construction from Arabic texts and solution. The framework consists of four main phases: pre-processing of corpus, concepts extraction by group sets of candidates into a unique set of concepts then validated by expert, concept relation exploration by combining linguistic, static and data mining techniques, and finally ontology building. The author discuss the important and characteristics of Arabic language. Also discuss some of the current issues and open questions of the ontology construction from Arabic text.

Mazari et. al. (Mazari, Aliane, & Alimazighi, 2012) develop an approach for automatic construction of ontology of domain for Arabic linguistics using statistical techniques. The initialization of the ontology is started manually by generic concepts retrieved from the ontology of General Ontology for Linguistic Descriptions "GOLD", a general ontology for descriptive linguistics. Constructing ontology includes: the formation of the domain corpus, the extraction of candidate terms and associated with the domain by "repeated segment", co-occurrence to link new term extracted to the ontology by hierarchical or non-hierarchical relations, and identify relations between terms by studying the context surrounding terms in small window. The method looks for lexico-syntactic elements to identify a relation between them. The relation uses "is-a" and "part-of". Test the approach using the Python programming language. The program gives the result in a marked file where each

line contains the co-occurring, their frequency and their co-frequency. Result file must be validated by an expert.

Mezghanni and Gargouri (Mezghanni & Gargouri, 2014) propose an approach for ontology learning from Arabic legal texts, the process consists of two main steps, corpus acquisition and ontology extraction process based on: first logical structure extraction these structures reflect the document's logical units hierarchy and its representation in a well-formed XML document. Second content text extraction containing three phases: linguistic, semantic and statistical. The combine and cooperate the various available sources as document content, document structure and external lexical resource.

Harrag et. al. (Fouzi Harrag, Abdulwahab Alothaim, Abdulaziz Abanmy, Faisal Alomaigan, & Alsalehi, 2013) build ontology for *Sahih Al-Bukhara* book and uses association rules to extract the ontology of prophetic narrations (Hadith). The ontology is divided into two principal parts. First part is related to the structure of *Sahih Al Bukhairi*. The second part is related to the global ontology that represents the main concepts of hadith as semantic relationships. They investigate the use of association rules to identify frequent item-sets over concepts that are related to Islamic jurisprudence from the *Sahih Al-Bukhari* documents by computing correspondence relations using the Apriori algorithm. The association rules express relations between classes of connected concepts in the *Sahih Al-Bukhari* collection. OWL can be used to explicitly represent the meaning of terms in vocabularies and the relationships between terms. The authors take four hadiths from *Sahih Al-Bukhari* as examples for illustrate the process, interest in the relations of type "is a part of". There are not comparison, results for experimental in this paper.

3.3.2 Automatic English Ontology Construction

Azevedo et. al. (Ribeiro de Azevedo et al., 2014) propose an approach based on ontology learning and natural language processing for automatic construction of expressive ontologies, specifically in OWL DL from English text. The architecture of the approach is composed of three modules: syntactic parsing module where this module uses Probabilistic Context-Free Grammars (PCFGs), semantic parsing module that perform terms extraction as noun then concatenation term that extracted

after that break the phrases and final relations extraction, OWL DL axioms module which finds axioms to prevent ambiguous interpretations. The approach can aid developers to start creating ontologies and results obtained through the experiments prove that they are sufficient to create ontologies with ALC expressivity. The approach constructs the expressive ontology correctly in more than half of the cases; and does not construct the expressive ontology in more than half of the cases. Analysing all the 120 sentences, the results obtained. In 75% of the sentences analyzed, the translator detected and created coherently the axioms, whereas; In 25%, the translator could not possibly solve in any way.

Hassanzadeh (Hassanzadeh, 2013) proposes a system for information extraction from plain text in form of RDF triples. The approach is capable of identifying grammatical structure (syntactic) of an input sentence and analyse its semantic to generate meaningful RDF triples of information, by Stanford and Senna tools for translating plain text documents into a machine-readable format which covers both syntactic/grammatical and semantic/conceptual information. For evaluation, compare the results obtained from proposed approach with the one obtained using FRED (a text to RDF convertor tool), and also with a set of triples created by a human. The results expressed a better representation of texts in most of the case studies, but not able to produce and cover all writing styles as many as triples possible from a text.

Nguyen et. al. (Nguyen, Nguyen, Ma, & Pham, 2011) develop a system that automatically builds ontology from Vietnamese texts using cascades of annotation based grammars. Gate is used to implement the system. The system includes two components: syntactic analysis and ontology construction, the syntactic analysis is to detect noun phrases and relation phrases from input documents, and then identify candidate phrases representing classes, individuals, relationships and properties, subsequently, the ontology extraction component uses Text2Onto (Cimiano & Völker, 2005) to generate the output ontology. Experiment results for classes, individuals, relationships and properties give f-measures as 67%, 67%, 52%, 71%.

Gantayat (Gantayat, 2011) presents a technique for automatically constructing ontology from a given lecture notes. This system extracts the concepts

using Term Frequency Inverse Document Frequency(TF-IDF) weighting scheme and then determines the associations among concepts using apriori algorithm. Evaluating the system is performed by comparing the results with the dependencies determined by an expert in the subject area.

Overall, these works reflect a growing interest on ontology construction on several areas and the importance of the ontology on representing basic terms, concepts and relations as well as knowledge in a certain domain. Some of the works reviewed above are dealing with semi-automatic learning ontology from text or web documents. However, there are few efforts concentrating on Arabic language and they depend on one direction to extract elements of ontology either as terms or relations. They did not tackle the taxonomic relationships in constructing ontology. We propose hybrid approach to automatically constructing Political News ontology in Arabic language, extract terms using machine learning, then extract taxonomic relations using syntactic and lexical patterns, and finally the output is a ontology based on taxonomic relations.

Chapter 4

Automatically Constructing Domain Ontology from Arabic Text

Chapter 4

Automatically Constructing Domain Ontology from Arabic Text

In this chapter, we present our approach to construct ontology from Arabic text by extracting taxonomic relations from documents annotation. Our approach will be used to extract entities and taxonomic relations to construct Arabic ontology from the domain "الأخبار السياسية" (Political News). We start with an overview of our approach then pre-processing stage, features extraction, terms extraction, taxonomic relations extraction, knowledge representation. We proceed towards an elaboration of each of the individual stages in the overall process.

4.1 Approach Overview

Our overall approach to construct ontology from text is divided into five main stage, these stages are shown in Figure (4.1): (i) pre-processing the text where a set of NLP processing is performed including a sentence detection, tokenization, normalization to prepare the documents to be input to next stage. They are implemented using the GATE framework. (ii) features extraction, where the main objective of this phase is to obtain the morphological and syntactic structure of each sentence in the corpus such as POS tagging and stemming. (iii) named entity recognition to terms extraction by machine learning and some rules to enhance entity recognition. (iv) taxonomic relations extraction between entity pairs to generate triple statements in subject-predicate-object format. This done using rules and patterns. (v) annotate terms and relations in the document to visualize taxonomic relations and their entities. Then transform the annotated text into ontological classes and relationships.

Next we elaborate these stages showing results of applying them throughout the approach.

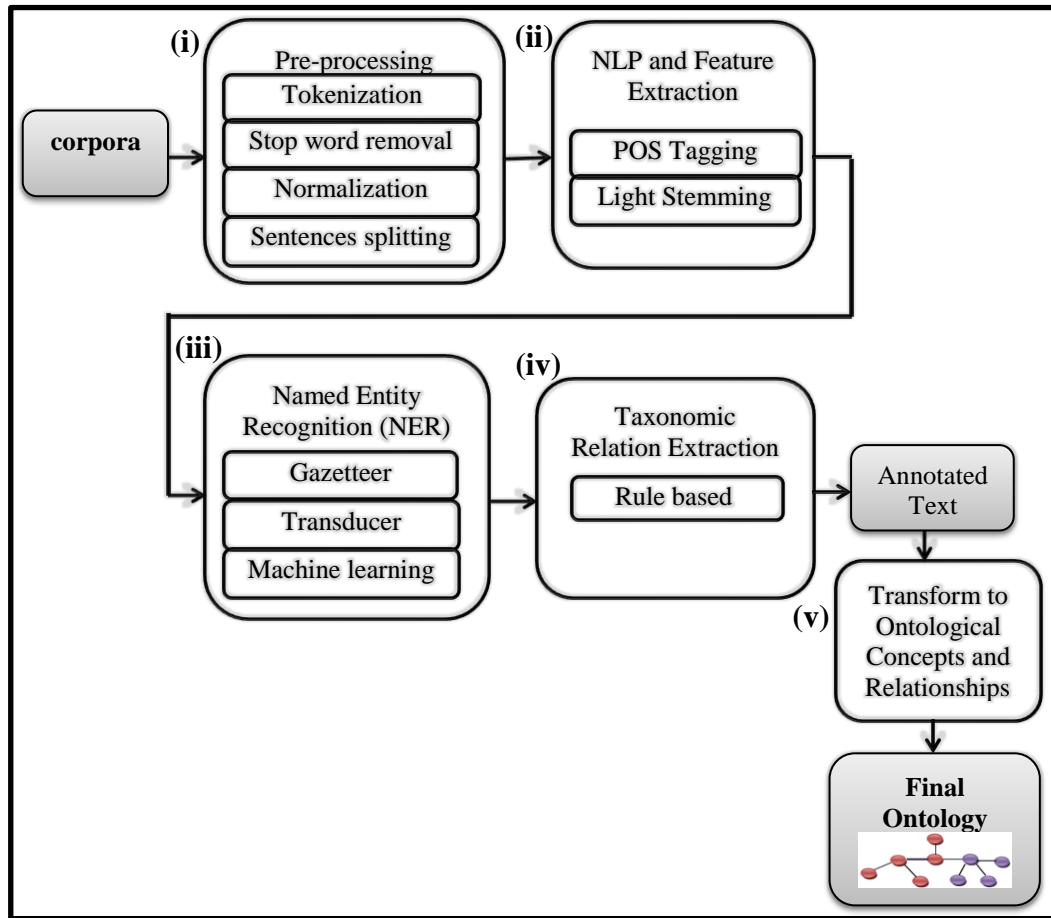


Figure (4.1): The Approach to Construct Ontology from Text

4.2 Pre-processing Stage

Pre-processing is one of the most important tasks and critical step in Natural Language Processing (NLP) and Information Retrieval (IR) which aims to prepare the documents to be input to next step of terms extraction. Pre-processing of the Arabic text is a challenging stage, it may impact positively or negatively on the accuracy of any information extraction system. Pre-processing step can contains many sub processes and each one has a specific function to prepare the data to be easily accessible representation of texts that is suitable to construct ontology. As shown in Figure (4.2) the proposed pre-processing focuses on the following steps:

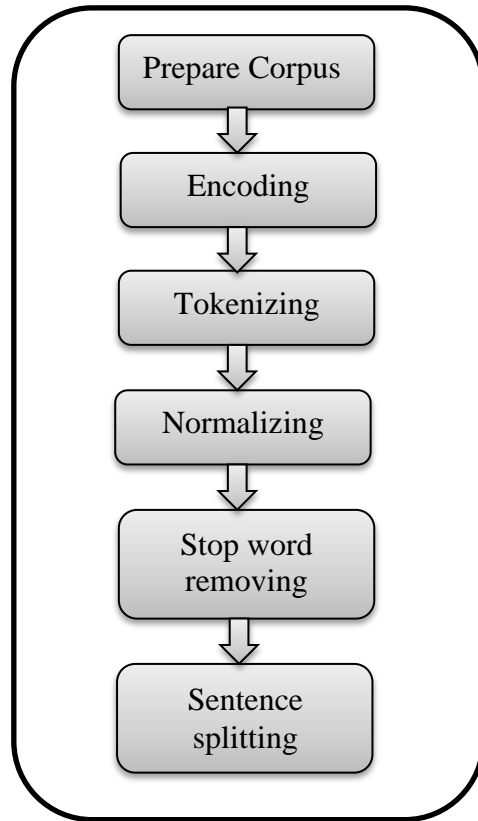


Figure (4.2): Pre-processing Stage

4.2.1 Preparing the Corpus

Preparing the corpus is one of the most important stages in the approach. The corpus is a collection of documents in one domain. We use these documents in the process of extracting taxonomic relationships to construct ontology. We collect nearly 3845 documents related to our Arabic ontology domain "الأخبار" (News). We collect these documents from bbc-arabic.com. We concentrate in the part of "الأخبار السياسية" (Political News) and build the ontology depending on it. All the documents collected are in plain text when we load them into Gate in order to facilitate the processing.

4.2.2 Encoding

This step deals with unifying the encoding to avoid character appearance problems and to standardize any dataset for future use. Also it is used in the normalization process.

Encoding is a numbering scheme that assigns each text character in a character set to a numeric value. A character set can include alphabetical characters, numbers, and other symbols. Different languages commonly consist of different sets of characters. So many different encoding standards exist to represent the character sets that are used in different languages. Windows CP-1256 encoding ("Unicode Windows 1256", 2015) is a code page used to write Arabic, Persian, and Urdu under Microsoft Windows, it encodes every abstract single letter of the basic Arabic alphabet, not glyphs. Another popular encoding is UTF-8 ("Unicode 8.0 Character Code Charts" 2015) which is a variable-width encoding that can represent every character in the Unicode character set. We choose UTF-8 for all text content by converting any content to UTF-8.

4.2.3 Tokenization

Tokenization is the procedure of analysing and splitting the input text into a number of tokens such as, number, word, space, symbol, etc. as shown in Figure (4.2). It is necessary step in our NER process, in machine learning for NER and for pattern extraction of taxonomic relations.

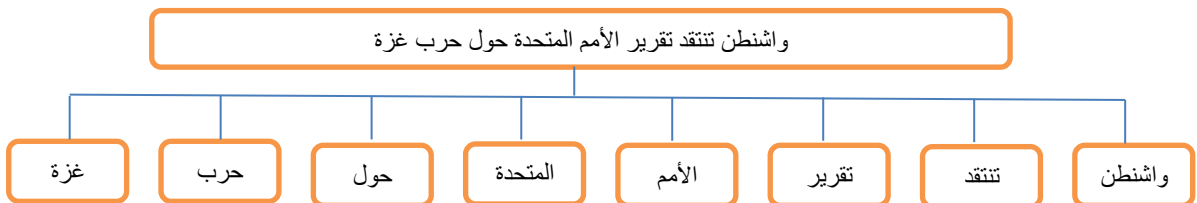


Figure (4.3): Tokenization Process

4.2.4 Normalization

Normalization replace different variations of a letter with a general form of the same letter, also it often removes punctuation, non-letters and diacritics (primarily weak vowels). The normalizing process (Almusaddar, 2014) depends on the Unicode number for every character to be used as the unique identifier for this character. Normalizing dataset and removing extra characters is very important. The normalization process we used is as follows:

1. Remove diacritics along with the short vowels, *shadda* and *sikkun*.
2. Remove all punctuation.
3. Use of regular expression "\\p {Punct}".
4. Remove numbers.
5. Remove non letters like special characters.
6. Replace َ, ِ, and ِ with َ.
7. Replace final ى with ي.
8. Replace final ة with ة.
9. Replacing the two final letters ى and ة with ئ.
10. Replacing the two final letters ي and ة with ئ.

4.2.5 Stop-Word Removal

Stop-word removal is a procedure of eliminating language words that do not carry any significance to a text or carry little meaning. The removal of the stop-word changes the document length and subsequently affects the weighting process. Also it can increase the efficiency of the indexing process as 30% to 50% of tokens in a large text collection can represent stop-words (El-Khair, 2006). Categories of stop-words cover adverbs, conditional pronouns, prepositions, and pronouns, transformers such as verbs, and letters, referral names and affixes such as prefixes, infixes, and postfixes. Table(4.1) lists some of Arabic stop-words.

The list of Arabic Stop-words ("Arabic Stop Words," 2013) will be used and updated by prevent remove some stop-word in documents. We eliminate any matching between stop- word list and my dataset words.

Table (4.1): Stop-Word List Sample.

ان	وكان	عليها	ومنذ	أما	عنه	وكانت
أن	تلك	الذي	اما	حول	وليس	هذا
إن	حتى	وتلك	وعلى	والذي	إما	دون
بعد	وحتى	كذلك	لكن	الذي	حين	اللاواتي
ضد	وهو	وكذلك	اللذان	ومن	لكنه	اللتان

4.2.6 Sentence Splitting

The sentence splitter groups the tokens in the text into sentences, based on tokens indicating a separation between sentences. Then input text is segmented into several sentences. Besides, the boundaries of the sentence can be classified by symbols such as end of line, punctuation and full stop. This process is significant for relations extraction where relations are often between two terms in a sentence. Consequently sentence splitter specify the boundary of relationship and prevent exceeding the range of statement to specify triple of domain, relation and range. As example

" وكان الرقيب إيهاب خطيب ينتمي إلى الطائفة الدرزية في إسرائيل ، حيث ان اسرة الرقيب إيهاب الخطيب تتكون من 7 افراد".

This statement consists of two sub-statement where the splitter is the comma "," and each statement has key elements to build ontology. We use splitting to specify the boundary of statement that contain taxonomic relations and terms pairs, then identify RDF triple, used this from Gate resource.

4.3 NLP and Features Extraction

In this stage we extract some features that are important to our approach. This features are considered as input for named entity recognition and taxonomic relations extraction. As shown in figure (4.4) features include POS and morphological analysing (light stemming).

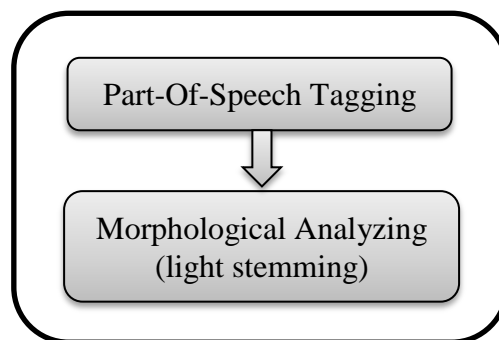


Figure (4.4): Features Extraction Stage

4.3.1 Part-Of-Speech Tagging

Part-Of-Speech (POS) Tagging involves identifying and adding parts of speech tags to text tokenized model. The POS tagger determines the syntactic category of each token, i.e. identifying nouns, verbs, adjectives and other parts of speech for each token. POS is encoded in capitalized abbreviations. For instance, syntactic categories with suffix VB are verbs, e.g., VBZ denotes a verb in third person singular present. Categories beginning with NN are nouns, such as a single proper noun NNP. Common syntactic categories are displayed in Table (4.2). We use to extract POS tagger Arabic Stanford POS tagger (Kristina Toutanova, 2003). It is the essential basic tools required in speech recognition, parsing, information retrieval and information extraction. The majority of the words in the text have more than one morphological analysis. The responsibility of POS tagger is assigning each word with the most suitable morphological tag.

The POS tagging is a step applied as a feature for each token, where we use this feature to NER by machine learning. POS attribute is the main feature to detect the NEs. We can use POS tag also as patterns to extract relationships from text (Maynard et al., 2001).

We use the Arabic Stanford POS tagger as plugin in GATE framework. It appear as category feature in the property of the token word in GATE. Use category feature as input to machine learning to extract terms from texts.

Table (4.2): Common Syntactic Categories

Category	Description
CC	Coordinating conjunction
CD	Cardinal number
IN	Preposition
JJ	Adjective
NN	Noun
NNP	Proper Noun
PP	Pronoun
RB	Adverb
VB	Verb, base form
VBZ	Verb, third person singular present

4.3.2 Morphological Analysis

Because different forms of a word that have a similar meaning; they relate to the same concept. Therefore, the stemmer component reduces the forms of the words. Stemming is deletion of prefix and suffix characters to identify the root word. Stemming aims to find the lexical root in natural language and one of the most important factors that affect the performance of information retrieval systems.

Light stemming is to find the representative indexing from of a word by removing affixes (Almusaddar, 2014). The main goal of light stemming is to get a better reduce the small size of the word to improve the information extraction and named entity recognition, by application of truncation of affixes. Light stemmer is not concerned with root extraction. For instance, for the token "المؤسسون" when we perform light stemming in this token the result is "مؤسس", suffix ="ون" and prefix ="ال". When the machine learning in NER deals with the word without stemming the same semantic words produce many variants which consume the size and the process time, in contrast stemming can improve the effectiveness of terms extraction and taxonomic relation extractions when building JAPE rules as patterns to extract lexical word, for example when extract lexical word "تتضمن" or "يتضمن" that is same word to represent taxonomic relations.

The light stemming algorithm was used from Almusaddar (Almusaddar, 2014), the algorithm is designed to improve Arabic light stemming in information retrieval systems. The algorithm consists of three stages to perform light stemming, normalizing using introduced a set of rules to be standardized, stop-word removal using introduced two different stop-word lists, the first one is intensive stop-word list for reducing the size of the index and ambiguous words, and the other is light stop-word list for better results with recall in information retrieval applications.

Improved light stemming by update a suffix rule, which is based on Larkey work, as this suffix is widely used, so it has been added to the list of the suffixes for better stemming of Arabic words, so the sequence of "ت", "ن" and "ا" characters are added to the enhanced suffixes list for best results, and introduce the use of Arabized words, 100 words manually collected, these words should not follow the stemming rules since they came to Arabic language from other languages.

Terms Extraction Stage

Named entity recognition (NER) is the task of identifying proper nouns in unstructured text (Nadeau & Sekine, 2007). The simplest and most reliable IE technology is NER. We propose a simple integration between lexical resources with some rule based system and machine-learning classifier for Arabic RER. A named entity (NE) is a word or phrase that contains the name of: person, location, an organization, dates, amounts of money, number, percent, nationality, product wars, substance or quantity. For example, the sentence "فرنسا تعتبر من ضمن الاتحاد الاوروبي" contains two named entities "فرنسا" is a location, "الاتحاد الاوروبي" is an organization. For our proposed terms extraction, we use Gazetteers as linguistic knowledge where they are able to detect complex entities, and then we enhance detected entities with rules. After the corpus is annotated (tagged), we use Machine Learning ML algorithms in order to determine NE tagging decisions from annotated texts that are used to generate statistical models for NE prediction. GATE offers a list of Gazetteers and allows the creation of user defined Gazetteers. Gazetteers can also store lists of keywords that can help identify some entities within documents. The gazetteer lists are compiled into finite state machines to be able to match the text tokens. Next we present steps to perform terms extraction as shown in Figure (4.5) that are lexical resources for NER, transducer, machine learning based NER.

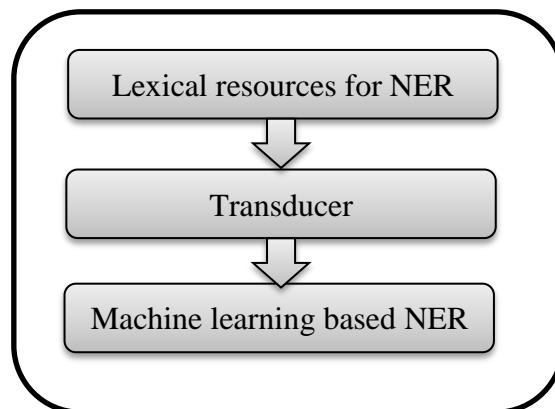


Figure (4.5): Terms Extraction Stage

4.3.3 Lexical Resources for NER

A primary linguistic resource and needed for NER is Gazetteer, which is a collection of predefined lists of typed entities. The role of the gazetteer is to identify entity names in the text based on lists. The Gazetteer lists used are plain text files, with one entry per line (Maynard et al., 2001). Each list represents a set of names, such as person name, location, organization, etc. Various linguistic resources are necessary in order to develop the proposed Arabic NER system with scope of 11 different categories of NEs. A summary for categories of named entities is shown in Table (4.3).

Table (4.3): Categories of Named Entity

No	Entity	No. of instance	No	Entity	No. of instance
1	Person	3491	6	Number	433
2	Location	1282	7	Nationality	459
3	Organization	281	8	Products Wars	132
4	Date	1001	9	Quantity	113
5	Money	105	10	Substance	71

Table (4.4) lists instances of location, person and organization category.

Table 4.4): Category Lists in Gazetteer

Location list	Person list	Organization list
لبنان	إحسان	وكالة المخابرات المركزية
سورية	ابو بكر	سي ان ان
الاردن	ابو مازن	الأمم المتحدة
مملكة البحرين	ابو عمار	مجلس الامن
دولة الكويت	أحمد	منظمة التجارة الدولية
الكويت	إدريس	الجامعة العربية

Creation of Gazetteer resource is done using Embedding GATE (GATE API). When using Gate API, it is easy to use external resources in Gate and control all Processing Resources (PRs) that represent primarily algorithmic, such as parsers, Gazetteer resource. Then creating the annotated documents.

4.3.4 Transducer

Text usually contains many kinds of names such as person names, company names, location names such as city and country, and lots of other names forming a specific domains. Rule-based NER is used to automatically locate and classify these names into predefined categories. For our purpose, we use rules and patterns to extract Arabic NEs from text using regular expressions. We depend on trigger word to extract person name, location name and organization name as shown in Table (4.5). After determining the trigger word we need to examine the word next to the trigger word it as a proper noun and not located in gazetteer lists.

Table (4.5): Trigger Word

Phrase	Trigger word	Named entity type
هيئة الامم المتحدة	هيئة	Organization
منظمة حقوق الإنسان	منظمة	Organization
شركة الغاز الطبيعي	شركة	Organization
جمهورية مصر	جمهورية	Location
مملكة البحرين	مملكة	Location
دولة العراق	دولة	Location
صرح الرئيس ياسر عرفات	الرئيس	Person
قال السيد محمد خالد	السيد	Person

For location recognition we collect keywords ("جمهورية", "مملكة", "دولة") as trigger to extract location such as countries. Where this word combined with location entities.

4.3.5 Machine Learning Based NER

ML-based NER systems take advantage of the ML algorithms in order to learn NE tagging decisions from annotated texts. The most common ML techniques used for NER are Supervised Learning (SL) techniques which represent the NER problem as a classification task and require the availability of large annotated datasets. Among the most common SL techniques utilized for NER are Support Vector Machines (SVM), Conditional Random Fields (CRF), Maximum Entropy (ME) and Decision Trees (Nadeau & Sekine, 2007).

ML-based component depends on two main aspects: feature engineering and selection of ML classifiers. Feature engineering involves the selection and extraction of classification features. ML classifier is used in the training, testing and prediction phases (Shaalán, 2014).

Exploring different types of features and arranging them in sets allow studying the effect of each feature set on the overall performance of the proposed system along different dimensions, including NE type and ML technique.

Machine learning component utilizes the ML techniques to generate a classification model for Arabic NER trained on annotated datasets by Gazetteer. The feature set is selected to develop the ML-based component, where the features that are used in machine learning are token, previous token, next token, POS tags, word length and stemmer.

Feature set is a major element in machine learning. Supervised machine learning systems cannot be directly trained on a corpus annotated with named entities. The corpus has to be transformed into a collection of instances. Usually instances are generated for consecutive tokens excluding punctuation marks. All instances that are used in the machine learning process are represented as vectors, each is composed of the class identifying a particular type of named entity and the list of features. (AbdelRahman et al., 2010).

We extract and build the list of features that should be effective in solving named entity classification tasks. All features can be grouped into two main categories: language independent and language dependent. Language independent features are very general, based only on the orthographic information where available in the corpus; language dependent features resort to external resources such as POS tagger and stemmer (Kapociute-Dzikiene, Nøklestad, Johannessen, & Krupavicius, 2013).

Below we present a list of all the features that are used in our approach.

Language independent (basic and orthographic) features:

- Current token (T): Current word.
- Number (T): Boolean indicator that determines if T is a number.
- Length (T): Numeral indicator that determines the length of T.

- Previous two Token --(T).
- Next two Token (T)++.

Language dependent features:

- POS (T): The POS tag of T (e.g. POS "احمد" = Noun).
- Light Stem (T): The light stem of T (e.g. Stem "الرئيسية" = "رئيس"). The stemmer eliminates inflectional ending (and some other suffixes) of the input word.
- Noun flag: A Boolean feature which is true if the part of speech tag is noun and false otherwise.

Class labels:

The corpus contains ten types of named entities for person names, location, organizations, dates, amounts of money, number, percent, nationality, product wars, substance and quantity.

4.4 Taxonomic Relation Extraction

We are interested in relations between entities, such as person, organization, and location. Rule-based information extraction uses specific rules that describe patterns to be matched. This step involves finding appropriate patterns that detect relations, where a particular relation can be automatically extracted by applying a set of structural patterns to identify that relation.

Extraction rules capture taxonomic relations by identifying specific lexical elements in a text, such as keywords, Although extraction rules can be defined following regular expression, like GATE's Java Annotation Patterns Engine (JAPE) (Cunningham et al., 2009). For our approach, we create rules to specify extraction patterns. These specially designed languages allow the creation of complex extraction rules through the manipulation of annotations. One of the most well-known sets of extraction rules are Hearst's extraction patterns (Hearst, 1992). Hearst has identified a small set of specific linguistic structures which are a combination of lexical and syntactical elements that represent a hyponymy relationship between two or more entities. A hyponymy relation between two entities NP0 and NP1 refer to membership relations in the form NP0 is a (kind of) NP1, where (kind of) is one of the taxonomic relations category such as "نوع من".

A relation is defined in the form of a tuple $t = (e_1; e_2; \dots; e_n)$ where the e_i are entities in a predefined relation R within document D . Most relation extraction systems focus on extracting binary relations.

In our work, we aimed at finding all binary relations without any restriction to relation classes. Our main goal is to detect a set of words that predicts relations between NEs. The methodology for taxonomic relations extraction depends on pattern-based semantic relations extraction frequently involving three main steps:

4.4.1 Defining the Semantic Taxonomic Relation Category

The relations are organized into categories or separate lists, depending on the type of relation to be extracted, we have in each category a set of linguistic pattern. This category can help to group relations into lists. The semantic taxonomic relations category can be on the following:

- **Hyponymy**

It is an important relation for structuring lexical terms. For example it defines the relationship between Jerusalem and Palestine. Given two lexical items c_i, c_j and their set of real-world referents S_1 and S_2 , respectively, c_i is a hyponym of c_j if and only if $S_1 \subseteq S_2$. For describing this relation, $\text{Hyponym}(S_1, S_2) \rightarrow S_i = \{c_1, c_2, \dots, c_n\}$ and $\forall (c_i \subseteq S_1 \wedge c_j \subseteq S_2 \rightarrow \text{Hypernym}) \rightarrow \text{Hyponym}(c_i, c_j)$. (Hearst, 1992) We use patterns to extract hyponymy relation "مقبلة على", "هي عاصمة".

- **Part-whole**

For describing relation $\text{partOf}(S_1, S_2) \rightarrow S_i = \{c_1, c_2, \dots, c_n\}$ and $S_2 \subseteq S_1$ and $\forall (c_i \subseteq S_1 \wedge c_j \subseteq S_2 \rightarrow \text{Part}) \rightarrow \text{PartOf}(c_i, c_j)$. Example for this relation PartOf ("الكويت", "مجلس التعاون الخليجي"). We can use pattern to extract part-whole relation "من مكونات" and "عضوا في".

- **Kind-of**

It one of taxonomic relations, to description the kind-of relation $\text{KindOf}(c_1, c_2) \rightarrow c_1 \in S_1$ and $c_2 \in S_1$). The pattern we use "نوع من" and "احد". Example for this relation kind-of("فاكهة", "تفاح").

- **Has-a**

For description relation $\text{Has-a}(S1, S2) \rightarrow S_i = \{c1, c2, \dots, cn\}$ and $S2 \subseteq S1$ and $\forall (c_i \subseteq S2 \rightarrow c_i \subseteq S1)$. Example for this relation Has-a ("الكعبة", "مكة"). The pattern we used to extract Has-a relation "تقع في" and "له".

- **Is-a**

To description this relation $\text{Is-a}(S1, S2) \rightarrow S_i = \{c1, c2, \dots, cn\}$ and $\forall (c_i \subseteq C \wedge c_j \subseteq C)$. Example for relation Is-a ("الملك سلمان", "خادم الحرمين"). Pattern we used to extract Is-a relation "هي", "هو".

- **Cause-Effect**

To description this relation $\text{Cause}(S1, S2) \rightarrow S_i = \{c1, c2, \dots, cn\}$ and $\forall (c_i \subseteq S1 \wedge c_j \subseteq S2 \rightarrow \text{Effect}) \rightarrow \text{Cause}(c_i, c_j)$. Example for relation Cause ("الاحتلال الإسرائيلي", "لانتفاضة"). Pattern we used to extract Cause relation "بسبب", "نتيجة".

4.4.2 Discovering the Actual Patterns

Once relations category is identified, the linguistic patterns expressing these relations between terms. We needs to be discovered the context surrounding these terms in a small window (three word for each side). From this context the method looks for lexical elements for identifying taxonomic relations between terms. The strategies of pattern-based approaches consist of compiling lists of reliable patterns that can immediately specify semantic relation types and use these lists to find new instances in texts. These taxonomic patterns are summarized in Table (4.6).

Table (4.6): Taxonomic Relationships

Category	Taxonomic Relations Patterns Example
Is-a	هو احد - هي إحدى - هي عاصمة - مقبلة على - هي - هو
Cause-Effect	بسبب - نتيجة
Part Whole	عضو في - يتكون من - ينقسم الي - يتألف من - ينتمي الي - من مكونات - من فصيلة.
Has-a	له - تقع في - موجوده في - ضم - حوى.
Kind of	نوع من - احد الانواع.

4.4.3 Searching for Instances of s Relation using Patterns

A pattern-based semantic relation would include a term A, a term B, and a linguistic unit expressing a semantic relation between term A and B. Searching instances of a semantic relation in texts using linguistic patterns can be implemented in different ways. We can use Jape rule to search instances of taxonomic relation using regular expression, where both A and B are unknown terms linked by a known relation, as for example, is-a(A,B).

4.5 Transforming to Ontological Elements and Knowledge Representation

Now that concepts and taxonomic relations have been identified, it is possible to produce an explicit representation in ontological form. The representation of the knowledge using instances extracted and annotated from text is important task in ontology construction. We formulate conceptual classes, instances and their relationships to represent these information using existing Resource Description Framework (RDF). The motivation behind RDF representation is it enables the possibility of complex querying on the extracted information. We represent knowledge as subject-predicate-object triples. Subjects are resources and extracted as concepts, predicates are taxonomic relations, and objects are resources and extracted as concepts.

4.6 Summary

In this chapter, we first presented an overview of our approach for automatically constructing domain ontology from Arabic text. Then we elaborated the stages of the approach which consist of: pre-processing stage, feature extraction stage, NER and terms extraction stage, taxonomic relation extraction stage and finally transformation of the annotated text to ontological elements and knowledge representation.

In the next chapter, we implement the approach on automatically constructing ontology in the Political News domain in Arabic language.

Chapter 5

Implementation

Chapter 5

Implementation

This chapter details the implementation of the approach presented in the previous chapter for constructing domain ontology. Firstly we state the tools and programs used to develop the proposed model and then we implement the stages of the approach starting from the first stage considered as the starting point to terms extraction and ontology learning. Then we perform term extraction that involves applying information extraction methods to extract terms about specific domain from text and identifying words that are candidates for concepts from texts. The final stage is extracting taxonomic relations between pieces of information in the underlying context implies rules and patterns to extract taxonomic relation that are used in ontology learning.

Building the ontology involves determining the domain and specifying suitable method to achieving terms and taxonomic relations extraction. This requires identifying the initial steps of the process and illustrating what each step involves. Building the ontology involves pre-processing, terms extraction and taxonomic relations extraction. Pre-processing stage involves applying document pre-processing techniques to allow for lexical and semantic analyses in the texts. This is achieved by applying a stop words removing, normalizing, tokenization step followed by a POS tagging and light stemming.

Terms and taxonomic relation extraction stage is important layers of the ontology learning layer cake (as detailed in Chapter 2). The aim of this stage is to extract concepts after pre-processing stage and then extract taxonomic relations between two concepts, where the research focus is to extract taxonomic relation from unstructured data sources. Syntactically and semantically analysed documents to extract concepts and taxonomic relations is done by input from pre-processing stage, the output of this stage is applying to transformation rules to automatically produce ontological taxonomic relations with concepts such as domain, relation, range.

This chapter is structured as follows: tools and programs, implement pre-processing stage, terms extraction, taxonomic relations extraction, transformation of annotated text into ontological elements.

5.1 Tools and Programs

To realize the proposed approach, we utilize the following tools and programs.

- GATE Developer 8.1 toolkit: GATE developer is open source software capable of solving almost any text processing problem. The resources used from GATE are tokenizer, sentences splitter, Gazetteers to NER, machine learning to NER. Also show output of each resources as annotated text. (Maynard et al., 2001)
- Stanford Arabic POS: The Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in Arabic language and assigns parts of speech to each word (Toutanova et all, 2003).
- Almusaddar algorithm: to perform light stemming and normalization, this system enhance the stemming process for Arabic text (Almusaddar, 2014).

5.2 Pre-processing

This stage aims to prepare documents to be input to the next step of terms extraction. Pre-processing contains several sub-steps to achieve accessible representations of texts that are suitable for constructing ontology. Pre-processing includes first dataset preparation such as encoding, stop-words removing, normalizing. Then performs the pre-processing as tokenization, sentence splitting, POS, light stemming. In this step we use GATE API to perform all necessary sub-steps.

5.2.1 Datasets

Most IR research was extended out in English and supported by the annual Text Retrieval Conferences (TREC) sponsored by NIST (the National Institute of Standards and Technology). NIST has collected large quantities of standard data (text collections, inquiries, and relevance judgments) so that IR researchers can compare their techniques on common datasets. For Arabic language pre-processing test, the dataset Open Source Arabic Corpora (OSAC) collected by Saad and Ashour (Saad & Ashour, 2010) is applied in our approach. The corpus used is BBC Arabic

corpus that described in Table (5.1) and collected from bbc arabic.com. It includes 4,763 text documents. Each text document belongs to 1 of 7 categories (Middle East News 2356, World News 1489, Business & Economy 296, Sports 219, International Press 49, Science & Technology 232, and Art & Culture 122). The corpus contains 1,860,786 (1.8M) words and 106,733 distinct keywords after stop-words removal. In our approach, we extract terms and taxonomic relations from the category (Middle East News, World News), so the domain we concentrate on is Political News.

Table (5.1): BBC Arabic Corpus Details

Number	Category	Number of text document
1.	Middle East News	2356
2.	World News	1489
3.	Business	296
4.	Science & Technology	232
5.	Sports	219
6.	Entertainments	122
7.	World Press	49
Total		4,763

5.2.2 Encoding

To make the dataset compatible with the suggested ontology construction, we deal with unified encoding to convert from various encoding systems like Windows Arabic encoding (CP1256) to the Unicode UTF-8 system. Any dataset will first be analysed to detect the type of the encoding and then convert it to UTF-8 encoding. In the Unicode standard, version 6.3, Arabic range is from 0600 to 06FF in decimal format.

5.2.3 Normalization

In normalization, we remove punctuation, non-letters and diacritics (primarily weak vowels). The normalization process (Almusaddar, 2014) depends on Unicode number for every character to be used as the unique identifier for this character. Normalizing dataset and removing extra characters is very important. The normalization process involves:

- Removing diacritics along with the short vowels, *shadda* and *sikkun*.
- Removing all punctuations.
- Using regular expression "\\p {Punct}".
- Removing numbers.
- Removing non letters like special characters.
- Replacing َ, ِ, and ِ with َ.
- Replacing final ى with ي.
- Replacing final ة with ة.
- Replacing the two final letters ى and ة with ى.
- Replacing the two final letters ي and ة with ى.

The normalization was used from (Almusaddar, 2014) which improves Arabic light stemming in information retrieval systems, which uses normalization as pre-processing to enhance Arabic light stemming.

5.2.4 Stop-Word Removal

Stop-word removal is a procedure of eliminating language parts that do not carry any significance to a text or carry little meaning. The list of Arabic Stop-words ("Arabic Stop Words," 2013) will be used and updated by prevent remove some stop-word in documents. We use Java code to remove any matching between stop-word list and my dataset words and then call Gate interface to show results. Some types of stop-words are adverbs, conditional pronouns, prepositions, pronouns, transformers (verbs, letters) and referral names.

5.2.5 Sentence Splitting

Input text is segmented into several sentences. Besides, the boundaries of the sentence can be classified by symbols such as, end of line, punctuation and full stop. This process is significant for relations extraction where relations often between two terms in a sentence. We perform the splitter by a GATE resource called sentences splitting, which is annotated with the type "Sentence". It has a feature "kind" with two possible values: "internal" for any combination of exclamation and question mark or one to four dots and "external" for a newline. The following code demonstrates how to use the GATE API to perform sentence splitting and calling splitter in processing resource.

```

System.out.print(".....Begin Sentence Splitter.....");
FeatureMap f_splitter = Factory.newFeatureMap();
ProcessingResource splitte = (ProcessingResource)
Factory.createResource("gate.creole.splitter.SentenceSplitter", f_splitter,
Factory.newFeatureMap());
controller.add(splitter);

```

Figure (5.1) illustrates texts annotation with sentences splitting, annotation set "Sentence" must be check to present each start and end statement in document.

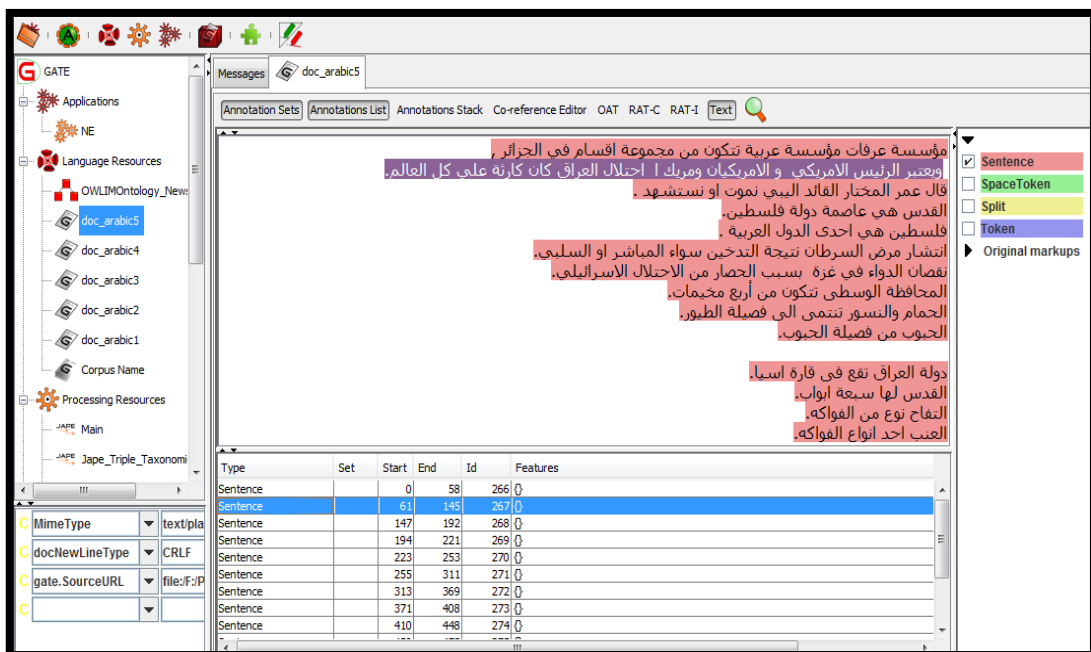


Figure (5.1): Sentence Splitting in Gate

5.2.6 Tokenization

Tokenization is the procedure of analysing and splitting the input text into a number of tokens such as number, word, space, symbol, etc. The function of a tokenizer we used in Gate API is ArabicTokeniser that breaks down text into segments or words.

The Java code using Gate API is used to perform tokenization by calling ArabicTokeniser as follows:

```

ProcessingResource arabicTokeniser =(ProcessingResource)
Factory.createResource("arabic.ArabicTokeniser");

```


The tokenizer is responsible for defining boundaries of a word. It is based mainly on the white spaces and punctuation marks as delimiters between words or major segments as in Figure (5.2). It shows a set of annotation set such as Token and Space Token. When checked annotation set Token then it annotated all tokens in text and shows start and end position properties for each token in the property region.

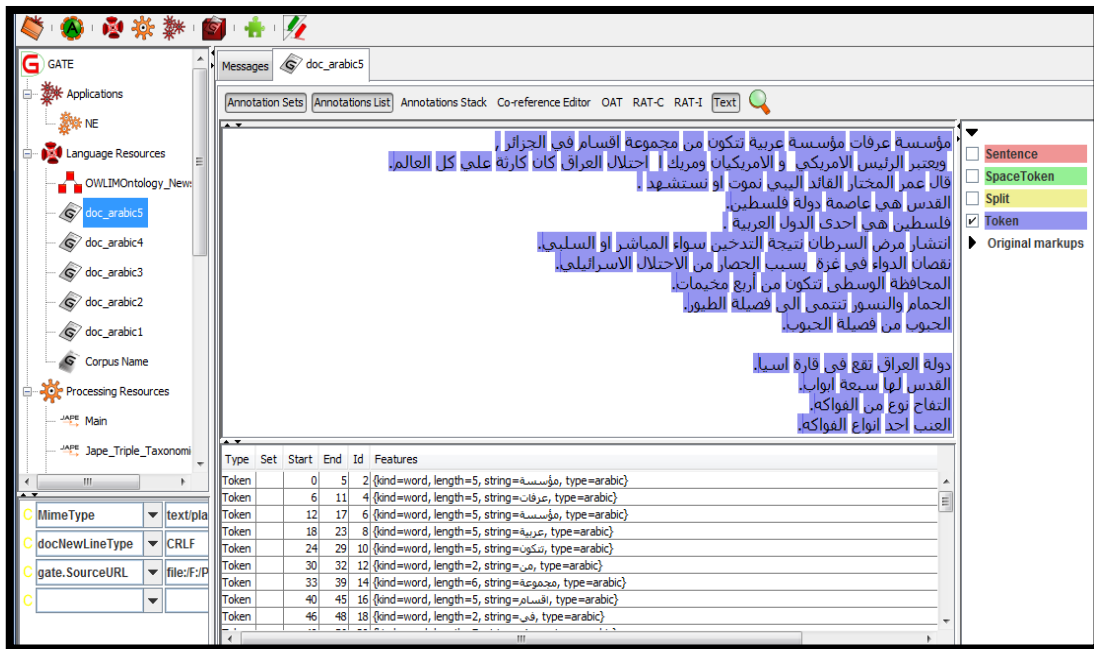


Figure (5.2): Tokenization Process in Gate

5.2.7 POS Tagging

POS tagging is main step before using named entity recognitions. It involves identifying and adding parts of speech tags to text tokenized model, i.e. identifying nouns, verbs, adjectives and other parts of speech for each token. The POS tagger used in GATE is the Hepple tagger which is a modified version of the Brill tagger (Cunningham et al., 2009), which produces a POS tag as an annotation on each word or symbol. The tagger in Gate uses a default lexicon and rule set which is not supported for Arabic language. So that we use the Arabic Stanford POS tagger (Toutanova et al., 2003) by loading the *Tagger_Stanford* plugin, and change the model file that is URL to Arabic Stanford parser model URL. Because we analysed the Arabic language, the POS tagger is added in the following example as a category feature in output.

type=Token; features={category=NNP, kind=word, length=7, string=Arabic};

The output from this phase looks like what is shown in Figure (5.3) for example for first token that show start and end boundary of token and also features show category of token is "NN" as noun, kind of token is word, and the word itself is "مؤسسة".

Type	Set	Start	End	Id	Features
Token		0	5	239	{category=NN, kind=word, length=5, string=مؤسسة, type=arabic}
Token		6	11	241	{category=NNP, kind=word, length=5, string=عرفات, type=arabic}
Token		12	17	243	{category=NN, kind=word, length=5, string=مؤسسة, type=arabic}
Token		18	23	245	{category=JJ, kind=word, length=5, string=عربية, type=arabic}
Token		24	29	247	{category=VBP, kind=word, length=5, string=تكون, type=arabic}
Token		30	32	249	{category=IN, kind=word, length=2, string=من, type=arabic}
Token		33	39	251	{category=NN, kind=word, length=6, string=مجموعة, type=arabic}
Token		40	45	253	{category=NN, kind=word, length=5, string=اقسام, type=arabic}
Token		46	48	255	{category=IN, kind=word, length=2, string=في, type=arabic}
Token		49	56	257	{category=DTNNP, kind=word, length=7, string=الجزائر, type=arabic}
Token		57	58	259	{category=PUNC, kind=punctuation, length=1, string=, type=arabic}
Token		61	67	263	{category=NNP, kind=word, length=6, string=ويعتبر, type=arabic}
Token		68	74	265	{category=DTNIN, kind=word, length=6, string=الرئيس, type=arabic}
Token		75	83	267	{category=DTJJ, kind=word, length=8, string=الامريكى, type=arabic}
Token		85	86	270	{category=CC, kind=word, length=1, string=و, type=arabic}

Figure (5.3): Part-of-Speech Features in Gate

5.2.8 Light Stemming

Stemming aims to find the lexical root in natural language and one of the most important factors that affect the performance of information retrieval systems and named entity recognition.

A Processing Resource (PR) in GATE is used to perform the stemming by applying the Porter Stemmer Algorithm (Porter, 1980) for English language, but there aren't stemming for Arabic language, so we use GATE API to add new feature called "Root" and assigned light stemming value to it. The light stemming algorithm we use in this work from Almusaddar algorithm (Almusaddar, 2014), the algorithm constructed to improve Arabic light stemming in information retrieval systems. We add new feature by the following JAPE rule:

ann.getFeatures().put("Root", Stem-Token);

While tokenizing the documents, a "Root" feature is applied to every token with the word light stem as its value as shown in Figure (5.4). It shows the Root feature for each token as first token "مؤسسة" there root feature is "مؤسس".

Type	Set	Start	End	Id	Features
Token		0	5	2	{Root=مؤسس, category=NN, kind=word, length=5, string=مؤسسة, type=arabic}
Token		6	11	4	{Root=عرف, category=NNP, kind=word, length=5, string=عرفات, type=arabic}
Token		12	17	6	{Root=مؤسس, category=NN, kind=word, length=5, string=مؤسسة, type=arabic}
Token		18	23	8	{Root=عرب, category=J, kind=word, length=5, string=عربية, type=arabic}
Token		24	29	10	{Root=تتك, category=VBP, kind=word, length=5, string=تتكون, type=arabic}
Token		30	32	12	{Root=من, category=IN, kind=word, length=2, string=من, type=arabic}
Token		33	39	14	{Root=مجموع, category=NN, kind=word, length=6, string=مجموعة, type=arabic}
Token		40	45	16	{Root=اقسام, category=NN, kind=word, length=5, string=اقسام, type=arabic}
Token		46	48	18	{Root=في, category=IN, kind=word, length=2, string=في, type=arabic}
Token		49	56	20	{Root=جزائر, category=DTNNP, kind=word, length=7, string=الجزائر, type=arabic}
Token		57	58	22	{Root=,, category=PUNC, kind=punctuation, length=1, string=,}
Token		61	67	26	{Root=يعتبر, category=NNP, kind=word, length=6, string=ويعتبر, type=arabic}
Token		68	74	28	{Root=رئيس, category=DTNN, kind=word, length=6, string=الرئيس, type=arabic}
Token		75	83	30	{Root=امريكى, category=DTJ, kind=word, length=8, string=الامريكى, type=arabic}

Figure (5.4): Stemming Features in Gate

5.3 Terms Extraction

Named Entity (NE) system extract all the names of people, locations, organization, dates, amounts of money, number, percent, nationality, product wars, substance, Quantity. GATE offers a list of gazetteers and allows the creation of user defined gazetteers. Gazetteers can also store lists of keywords that can help identify some entities within documents. The gazetteer lists are compiled into finite state machines to be able to match the text tokens. In this stage we extract terms by two step first lexical resource to identify entity names in the text based on lists, and machine learning based named entity recognition to learn NE tagging decisions from annotated texts.

5.3.1 Lexical Resources

The primary linguistic resource we use to extract named entities are the Gazetteer lists, which are a collection of predefined lists of typed entities. The predefined lists used are plain text files with one entry per line (Cunningham et al., 2009). Each list represents a set of names categorized into 10 lists such as person names, locations, organizations, nationality, product wars, substance, quantity, dates, money, numbers.

An index file (lists.def) is used to access these lists. For each list, a major type is specified and, optionally, a minor type. It is also possible to include a language in the same way (fourth column), where lists for different languages are

used, By default, the Gazetteer PR creates a Lookup annotation from every Gazetteer entry it finds in the text. In the example below, the first column refers to the list name, the second column refers to the major type, the third which is optional refers to the minor type, and the fourth which is also optional refers to language type.

city.lst:location:city:Arabic

months.lst:Date:month:arabic

organisations.lst:organisation::Arabic

We collect the lexical resource Gazetteer from the default ArabicGazetteer found in GATE, and additionally to enhance lexical resources we collected on external resource called ANERgazet ("ANERGazet," 2008). ANERgazet contains three categories which are person, organization and location. Using the Gazetteer resource is done by Embedded GATE (GATE API). When using GATE API, it is easy to use external resources in GATE and control all PR, and then creating the annotated documents. For example to create Arabic gazetteer as bellow.

ProcessingResource arabicGazetteer

= (ProcessingResource) Factory.createResource("arabic.ArabicGazetteer");

This code results as shown in Figure (5.5) in the annotated tokens when checked "Lookup" annotation set the NEs in Gazetteer is annotated on text. In the features for token "عرفات" there are majorType "person".

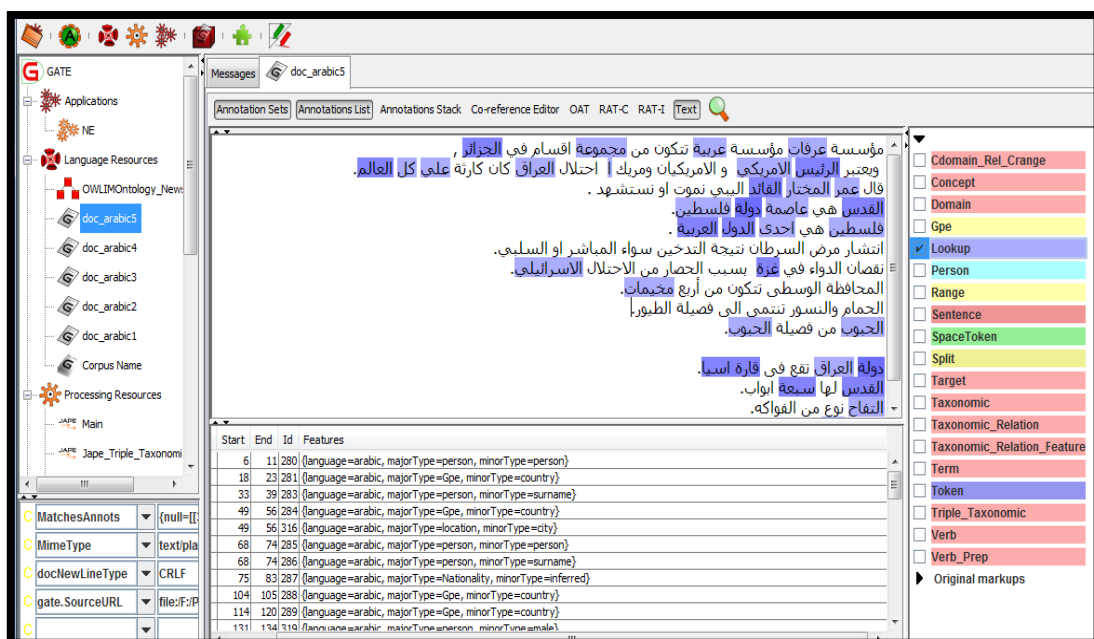


Figure (5.5): Gazetteer Resource in GATE

5.3.2 Machine Learning Based NER

Supervised Learning (SL) techniques is used for NER which represent NER as a classification task and require the availability of large annotated datasets. We use Support Vector Machines (SVM) as a SL technique.

ML-based component depends on two main aspects: feature engineering and selection of ML classifier. Feature engineering involves the selection and extraction of classification features. ML classifier is used in the training, testing and prediction phases (Shaalán, 2014). The ML component utilizes the two techniques to generate the classification model for Arabic NER trained on annotated datasets by the selected Gazetteer, (section 5.3.2).

NER is the task of detecting and classifying proper names within texts into predefined types, such as person, location and organization names. The feature set is selected to develop the ML-based component, where the features used in machine learning include token, previous token, next token, POS tags, word length and stemmer.

There are many tools available for developing and evaluating Arabic NER systems and machine learning. These tools also offer many features for the experiments. According to their functionality, we select one of the machine learning PRs available in GATE which is Batch Learning PR as part of the Learning plugin. It is specifically targeted at NLP tasks including text classification, named entity recognition. It integrates LibSVM for improved speed, along with the PAUM algorithm, offering competitive performance and speed (Cunningham et al., 2009). The features we use in ML are extracted from pre-processing stage and they are as follows:

- Current token (T): Current token of the annotated text.
- Number (T): Boolean indicator that determines if T is a number.
- Length (T): Numeral indicator that determines the length of T based on length feature from properties of annotated text.
- Previous two Token --(T).
- Next two Token (T)++.
- POS (T): The POS tagger of T, using Stanford parser for Arabic language.

- Light Stem (T): To eliminate inflectional ending of the input word.
- Noun flag: A Boolean feature which is true if the part of speech tag is Noun and false otherwise.
- Named Entities: Class labels, there are ten types of named entities for person names, location, organizations, dates, amounts of money, number, percent, nationality, product wars, substance and quantity.

Batch learning is the latest machine learning PR in GATE. We used SVM in the implementation of NER. The PR handles training and application of an ML model. For training the ML model we perform three steps: First annotate some training documents with the labels or classes. Second, perform pre-process on documents to obtain linguistic features for the learning, where this feature appears in annotations set and in features of the annotations. Finally, create a configuration file for setting the ML PR, in configuration file we select the learning algorithm and define the features that we selected to use in learning. The configuration parameters are set through external XML file. The XML file contains both the configuration parameters of the Batch Learning PR itself and of the attributes we selected. The XML file is specified when creating a new Batch Learning PR. The complete configuration of ML is founded in Appendix F.

5.4 Taxonomic Relations Extraction

Taxonomic relations extraction involves applying an appropriate rule based pattern-matching. This enable searching for and annotates relations and concepts related to the input token and creates the corresponding ontological elements.

Patterns are discovered by querying the underlying text using JAPE rules that produce a sequence of words that involve taxonomic relations between terms.

Taxonomic relations are organized into separate lists, depending on the type of relation to be extracted, we have in each category set of linguistic pattern. This category helps to group relations into lists. The semantic taxonomic relations categories are: Is-a, Cause-Effect, Part-whole, Has-a, Kind-of. As found in (Al Zamil & Al-Radaideh, 2014) (Mazari, Aliane, & Alimazighi, 2012).

Table (5.2) illustrates the patterns per category used in the JAPE rules for extracting taxonomic relations. These patterns include subcategories that exist between pairs of entities.

Table (5.2): Rules of Taxonomic Relations

Category	Patterns
<i>Is-a</i>	<pre>{Token.string == "هو" }({Token.string == "احد" })? {Token.string == "هي" }({Token.string == "حدى" })? {Token.string == "مقبلة" }({Token.string == "على" })? {Token.string == "هي" }({Token.string == "عاصمة" })? {Token.string == "هو" } {Token.string == "هي" } {Token.string == "هم" } {Token.string == "هما" } {Token.string == "هؤلاء" } {Token.string == "هن" }</pre>
<i>Cause-Effect</i>	<pre>{Token.string == "بسبب" } {Token.string == "يسبب" } {Token.string == "نتيجة" }</pre>
<i>Part-whole</i>	<pre>{Token.string == "عضو" } {Token.string == "فى" } ({Token.string == "تتكون" }) {Token.string == "يتكون" } {Token.string == "من" } ({Token.string == "تنقسم" }) {Token.string == "ينقسم" } {Token.string == "الى" } ({Token.string == "تتألف" }) {Token.string == "يتألف" } {Token.string == "من" } ({Token.string == "تنتمى" }) {Token.string == "ينتمى" } {Token.string == "الى" } {Token.string == "من" } {Token.string == "مكونات" } {Token.string == "من" } {Token.string == "فصيلة" }</pre>
<i>Kind_of</i>	<pre>{Token.string == "نوع" } {Token.string == "من" } {Token.string == "احد" } {Token.string == "انواع" }</pre>
<i>Has-a</i>	<pre>{Token.string == "له" } {Token.string == "لها" } ({Token.string == "تقع" }) {Token.string == "يقع" } {Token.string == "فى" } ({Token.string == "موجود" }) {Token.string == "موجوده" } {Token.string == "فى" } ({Token.Root == "ضم" }) {Token.Root == "حوى" }</pre>

To build the pattern for regular expression we use JAPE Transducers (Thakker et al., 2009). These transducers are developed to perform rule-based pattern extraction. A JAPE rule consists of two parts, the left hand side (LHS) and the right hand side (RHS). The LHS of the rule identifies the patterns to be matched based on information generated by the previous steps (tokenization and POS tagging). The RHS identifies the annotation set to be created for the text that matches the pattern on the LHS. The result of executing this JAPE rule on the input files is that each token that matches the pattern is annotated with a concept annotation. An example of the

"is-a" pattern used in a JAPE rule to extract taxonomic relation is shown in Figure (5.6).

```

phase: Taxonomic_Relation
Input: Token
options: control = appelt
Rule: Is_a
(
  {Token.string == "هو" }({Token.string == "احد" })? |
  {Token.string == "هي" }({Token.string == "احدى" })? |
  {Token.string == "مقابلة" }({Token.string == "على" })? |
  {Token.string == "هي" }({Token.string == "عاصمة" })? |
  {Token.string == "هو" } | {Token.string == "هي" } |
  {Token.string == "هم" } | {Token.string == "هما" } |
  {Token.string == "هؤلاء" } | {Token.string == "هن" }
):mention
-->
:mention{
  gate.AnnotationSet predi = (gate.AnnotationSet) bindings.get("mention");
  gate.FeatureMap features = Factory.newFeatureMap();
  features.put("rule", "Is_a");
  outputAS.add(predi.firstNode().predi.lastNode(), "Taxonomic Relation".features); }

```

Figure (5.6): JAPE Rule for Taxonomic Relation Creation " is-a"

Appendix C includes all the rules used to extract the taxonomic relations.

After extracting taxonomic relations consisting of class-subclass relationships and properties-sub-properties relationships, we build triple statements each of which represents a single fact. Each triple statements consists of three fixed components of the form Subject-Predicate-Object, and this order should never be changed, where the subject and object are concepts we extract before. The predicate property is a name of a relation that connects these two concepts. The JAPE rule in Figure (5.7) is created to extract new triple statement for the whale annotated text and considering all the taxonomic categories and their patterns of Table (5.2) and JAPE rule that created for this patterns in Figure (5.6).

For example to illustrate JAPE rule in Figure (5.7), the code consists of two side. LHS identifies the patterns in the form of Subject-Predicate-Object. The subject

and object annotated with "Concept" annotation set. These concept may be simple or compound words as sequence of two or three concepts for maximum, such as "منظمة الصحة العالمية" consists of three word and "شركة الاتصالات" consists of two word. The predicate annotated between two concept with annotation set called "Taxonomic_Relation_Feature". RHS identifies the annotation set to be created for the output of matches pattern on the LHS that is "Domain_Rel_Range", "Domain" and "Range" annotation sets.

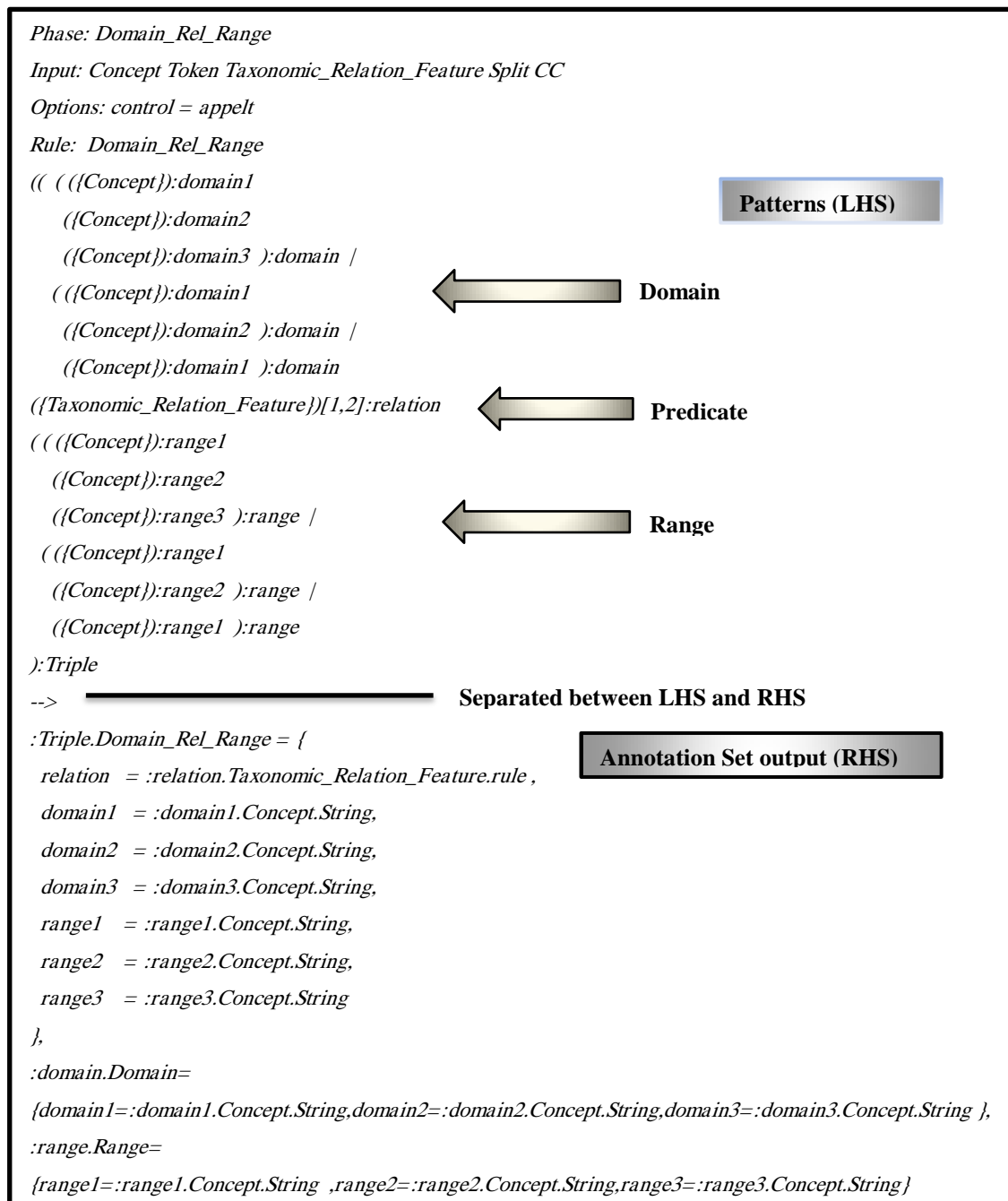


Figure (5.7): JAPE Rule for Building Triple Statements

5.5 Transformation of Annotated Text into Ontological Elements

This phase involves constructing a set of Transformation rules, which are used to identify an appropriate ontological elements from the extracted patterns. Ontological classes and relations can be automatically extracted using the previously NER and identified patterns to represent a particular concepts and relations. After determine the terms from document and based on Hierarchical of predefined lists (Gazetteer) for this term, that contain of list name, major type attribute and minor type attribute, so we transform this structure to classes and subclasses. In transformation phase the gazetteer list name such as Location list transformed to Location class, and major type of Location list such as (Arabic city, Arabic country , world city , world country , places, sea and island , facility) transformed to subclasses. Table (5.3) show and explain how mapping between taxonomic relations and OWL/RDF.

Table (5.3): Transformation of annotated text into ontological elements

Category	Patterns		
Triple Statement	<i>Domain</i>	<i>Predicate</i>	<i>Range</i>
Annotation Set	<i>Concept</i>	<i>Taxonomic Relation</i>	<i>Concept</i>
Annotation Feature	<i>MajorType-MinorType</i>	<i>TaxonomicRelationString</i>	<i>MajorType-MinorType</i>
Ontological Element	<i>Class-Subclass</i>	<i>Object Property</i>	<i>Class-Subclass</i>
	<i>Location-Country</i>	<i>Is-a</i>	<i>Country-City</i>
URI	"http://example.com/classes#" + Country	"http://gate.ac.uk/classes#" + هي_عاصمة	"http://example.com/classes#" + + City
Example	فلسطين	هي عاصمة	القدس

We develop another JAPE rule as shown in Figure (5.8) to find annotated concepts and relations in the text and create ontological concepts and resources accordingly. The ontology is created using the GATE OWLIM API.

```

Phase: Transform
Input: Cdomain_Rel_Crange //Domain_Rel_Range
Options: control = first //appelt

Rule: Transform
(!Cdomain_Rel_Crange):relationIden
-->
:relationIden{
    Annotation theInstance = (Annotation)relationIdenAnnots.iterator().next();
    String domain = theInstance.getFeatures().get("domain1").toString();
    String range = theInstance.getFeatures().get("range1").toString();
    String Rel = theInstance.getFeatures().get("relation_String").toString();
    // Create URI for domain and range.
    gate.creole.ontology.OURI domclassURI = ontology.createOURI("http://example.com/classes#" + domain);
    gate.creole.ontology.OURI rngclassURI = ontology.createOURI("http://example.com/classes#" + range);

    //Add domain and range concept to ontology
    gate.creole.ontology.OClass Domain = ontology.addOClass(domclassURI);
    gate.creole.ontology.OClass Range = ontology.addOClass(rngclassURI);
    .....
}

```

Figure (5.8): JAPE Rule to Create Ontological Concepts and Resources

The result of executing the rule in Figure (5.8) is an ontological classes-subclasses relations as shown in Figure (5.9). Annotated concepts and taxonomic relations on the text are transformed into ontological classes sub-classes relations.

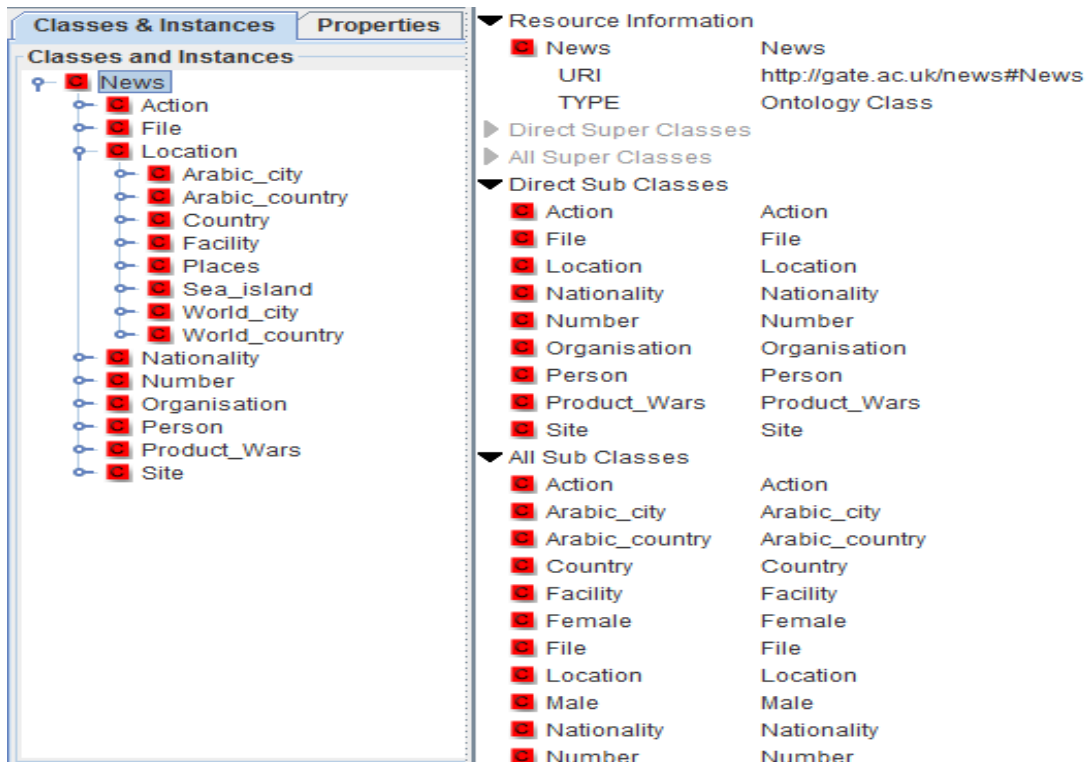


Figure (5.9): Classes and Subclasses for Political News Ontology

The second result of executing the rule in Figure (5.8) is an ontological properties relations as shown in Figure (5.10). Annotated taxonomic relations between concepts on the text are transformed into ontological properties for taxonomic relations. The triple statement extracted using patterns that contain three element first, terms as subject, second taxonomic relations as predicate, third terms as subject. To reflect this pattern to real example such as "القدس هي عاصمة فلسطين". The terms "القدس" is instance of location class and also instance of subclass of Arabic city. The taxonomic relation "هي عاصمة" is property located between two terms. The terms "فلسطين" is instance of location class and also instance of subclass of Arabic country.

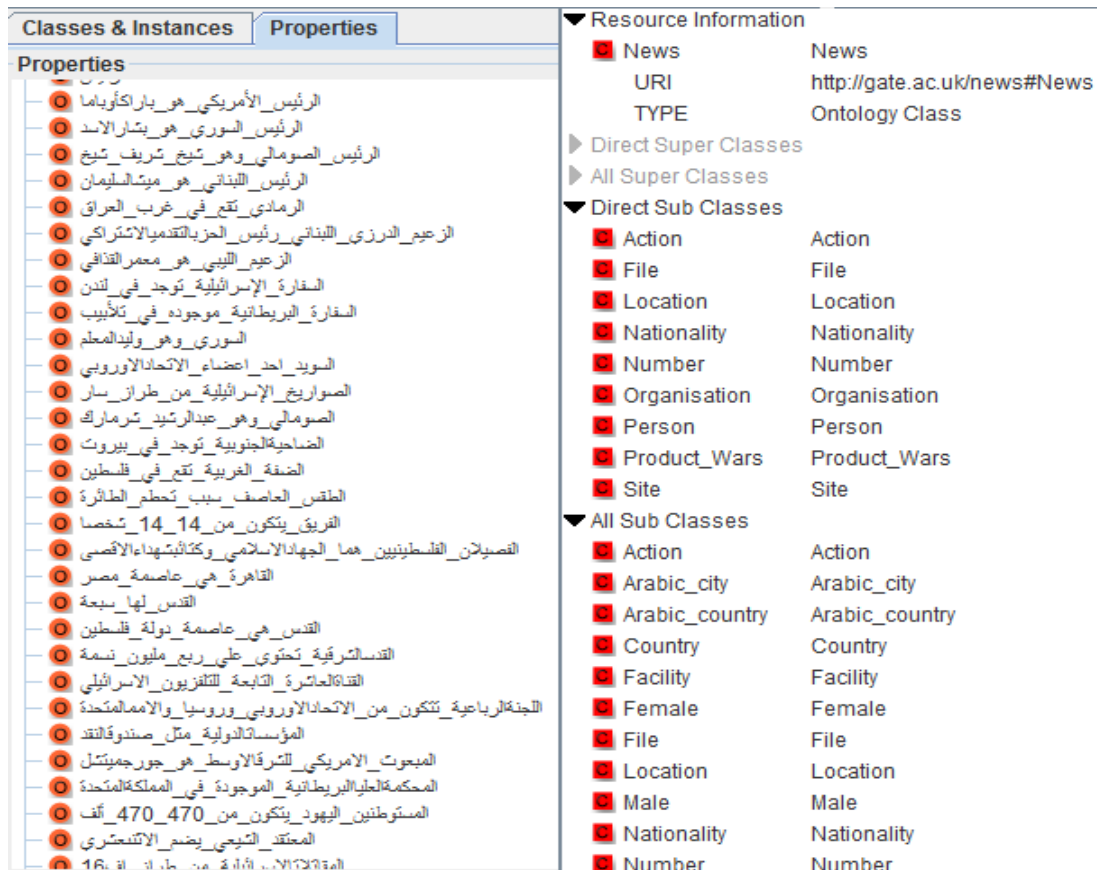


Figure (5.10): Ontological Properties for Taxonomic Relations

5.6 Summary

In this chapter, we presented details of implementations of the approach for the automatic construction of Political News ontology from texts. We started from tools and programs utilized, then the stages as specified in the approach consisting of: perform pre-processing, extracting terms and performing annotations, extracting taxonomic relations and annotating them in GATE framework, and finally transforming the annotated text to ontological elements as the target knowledge representation using a specified JAPE rule.

In the next chapter, we analyse the experimental result then evaluate the performance of the proposed approach.

Chapter 6

Experimental Results and Evaluation

Chapter 6

Experimental Results and Evaluation

In this chapter, the experimental results are present and analyse to provide evidence that approach can identify named entities, extract taxonomic relations, and construct ontology from Arabic text. In addition, the performance of the proposed approach in extracting taxonomic relations between terms and then automatic construct ontology are evaluate then inconsistent classes in the ontology are check. Finally, The results are visualize in a graphical representation to provide views of the final ontology as the final and desired results of the proposed approach.

6.1 Experimental Setup

To perform the experiments, tools used in experiments are describe to execute the proposed approach presented in Chapter 4 (including the experimentation), used the following tools:

GATE Tool: Used for natural language processing techniques in our approach to conduct experiments for extracting the terms and taxonomic relations from Arabic unstructured text, and construct the ontology.

Protégé Tool: Protégé is a free, open-source platform to construct domain models and knowledge-based applications with ontologies. It is used to check the consistency of the construction ontology and show Individuals, Properties, and Classes.

6.2 Arabic News documents Corpus

The corpus we used for training and testing is the Middle East News a BBC Arabic corpus from Open Source Arabic Corpora (OSAC) (Saad & Ashour, 2010).

The dataset is divided into two sets: The first dataset contains 700 documents used as a training phase in order to build rule-based approaches and modifies the Gazetteer lists. The second dataset contains 50 documents which is used by our system to test extracting ontology from text. As presented in Chapter 4, we perform all text pre-processing steps on the corpus; including encoding, sentences splitting,

tokenizing string into words and normalizing process to initialize the text. These all are performed based on GATE tools.

We create two annotations sets in GATE as shown in Figure (6.1), where the first annotation set (System Result, Concept) is used to extract terms and taxonomic relations automatically. The second annotation set (Domain-Expert-Result, Concept) is used by human experts to identify terms and taxonomic relations manually from documents selected randomly from the corpus. Consequently these results were used to calculate Precision, Recall and F-measure as described in Section 2.8 via using Annotation diff tool.

Type	Set	Start	End	Id	Features
Concept	Domain_Expert_Result	42	55	3796	{}
Concept		43	55	3660	{String=الشرق الأوسط, class=http://gate.ac.uk/news#Location, classM=http://gate.ac.uk/news#Places,
Concept		66	73	3661	{String=بريطاني, class=http://gate.ac.uk/news#Nationality, classM=http://gate.ac.uk/news#Nationality,
Concept	Domain_Expert_Result	66	73	3798	{}
Concept		74	78	3662	{String=جديد, class=http://gate.ac.uk/news#Person, classM=http://gate.ac.uk/news#Surname, classe,
Concept		88	94	3663	{String=البصرة, class=http://gate.ac.uk/news#Location, classM=http://gate.ac.uk/news#Arabic_city, c
Concept	Domain_Expert_Result	88	94	3799	{}

Figure (6.1): Annotation Set from System and Domain Expert

6.3 Data Pre-processing Results

GATE API tools has collection of operation that are suitable for NLP. There are many of pre-processing techniques such as stop word removing, document normalization, tokenization, sentences splitting and others. Before used this resource, the document reset resource is put to be reset documents to its original state by removing all the annotation sets. This technique help me to be easily accessible representation of texts that is suitable for the construct ontology method. For more details, show pre-processing methods used in our system as Figure (6.20).






Selected Processing resources		
!	Name	Type
	Document Reset PR_0004C	Document Reset PR
	ANNIE Sentence Splitter_00051	ANNIE Sentence Splitter
	Arabic Tokeniser_0004D	Arabic Tokeniser
	Stanford POS Tagger_00054	Stanford POS Tagger
	Add Root Feature	JAPE Transducer

Figure (6.2): Set of Processing Resources for Pre-processing Stage

6.4 Terms Extraction Result

GATE offers a list of resources for Named Entity Recognition, we create own pipelines special for Arabic language and made up a chain of Processing Resources.

In this thesis we create a new pipelines to handle our Arabic Political News corpus as described in Chapter 4. Processing resources is:

Gazetteer: Is a list build Name Entity Recognition (NER) describe in section 4.4.1 . we used gazetteer to identify typed entities.

Arabic Main Grammar: In this Processing Resources use Java Annotation Patterns Engine (JAPE) to execute regular expression and patterns base on rules, we build many JAPE rules to extract terms. The complete implementation of terms extraction is listed founded in Appendix A.

Arabic OrthoMatcher: The Arabic orthoMatcher Processing Resources detects orthographic coreference between named entities in the text. OrthoMatcher also can used to improve the name classification process by classifying unknown proper name. e.g. " باسل سلامة " and " باسل " usually refer to the same person name.

Machine Learning: Machine learning utilize techniques to generate a classification model for Arabic NER trained on annotated datasets by Gazetteer list and Arabic main grammar. Figure 6.3 Show the sample results of using Arabic Named Entity Recognition in GATE tool.

! Name	Type
Arabic Gazetteer_00055	Arabic Gazetteer
Arabic Inferred Gazetteer_00056	Arabic Inferred Gazetteer
Arabic Main Grammar_00057	Arabic Main Grammar
Arabic OrthoMatcher_00058	Arabic OrthoMatcher
Batch Learning PR_0005B	Batch Learning PR

Figure (6.3): Name Entity Extraction

To represent information about the text, and display various information about the texts being processed, Annotation sets and Annotation list features in GATE is used. However, different processing module such as tokenizer, POS tagging, Stemming property and NE transducer running over text, represent as show in Figure (6.4) using annotations features.

Type	Set	Start	End	Id	Features
Concept		13	3660		{String=الشرق الأوسط, class=http://gate.ac.uk/news#Location, classM=http://gate.ac.uk/news#Places, classes=مكان, majorType=Location, minorType=}
Concept		31	3956		{}
Concept		36	3681		{String=جديد, class=http://gate.ac.uk/news#Person, classM=http://gate.ac.uk/news#Surname, classes=شخص, majorType=Person, minorType=Sur}
Concept		53	3682		{String=البحر, class=http://gate.ac.uk/news#Location, classM=http://gate.ac.uk/news#Arabic_city, classes=مكان, majorType=Location, minorType=}
Concept		62	3954		{}
Concept		94	3685		{String=وزير الدفاع, class=http://gate.ac.uk/news#Person, classM=http://gate.ac.uk/news#Surname, classes=شخص, majorType=Person, minorType=}
Concept		104	3686		{String=البريطاني, class=http://gate.ac.uk/news#Location, classM=http://gate.ac.uk/news#Country, classes=مكان, majorType=Location, minorType=}
Concept		129	133	3687	{String=جديد, class=http://gate.ac.uk/news#Person, classM=http://gate.ac.uk/news#Surname, classes=شخص, majorType=Person, minorType=Sur}
Concept		145	147	3689	{String=20, class=http://gate.ac.uk/news#Number, classM=http://gate.ac.uk/news#Ordinal, classes=رقم, majorType=Number, minorType=Ordinal}
Concept		150	156	3690	{String=عراقيا, class=http://gate.ac.uk/news#Location, classM=http://gate.ac.uk/news#Country, classes=مكان, majorType=Location, minorType=Cou}

Figure (6.4): Sample of Name Entity Annotation

6.5 Taxonomic Relations Extraction Result

The next step in this experiment is to identify taxonomic relationships between terms we extracted first, and then discover triple of domain, relation and range. The algorithm was implemented using JAPE rule as regular expression and patterns to extract taxonomic relation between terms at least one NE, and at most four NE. The taxonomic relations is represented and displayed by annotation features

in GATE. Figure (6.5) Shows the sample results of taxonomic relations extraction from Political News document in GATE tool.

The screenshot shows the GATE tool interface. The main window displays Arabic text with highlighted segments. A right-hand sidebar lists various taxonomic relations. A table at the bottom shows the extracted relations.

Type	Set	Start	End	Id	Features
Cdomain_Rel_Crange		43	76	3742	{domain1=Location, domain1_Instance=الاستيطان, domain1_minor=Facility, range1=Location, range1_Instance

Figure (6.5): Sample of Taxonomic Relations Extraction

6.6 Ontology visualizer and Language Presentation

In this phase, we utilize Visual Resources (VRs) in GATE to present the constructed ontology visually. Tree visualization is considered useful for allowing the results of the relationship between terms to be more readable. Each term is represents as a node in the tree and a link is shown between two terms using taxonomic relations. Figure (6.6) shows classes and sub-classes relationships in the Political News ontology.

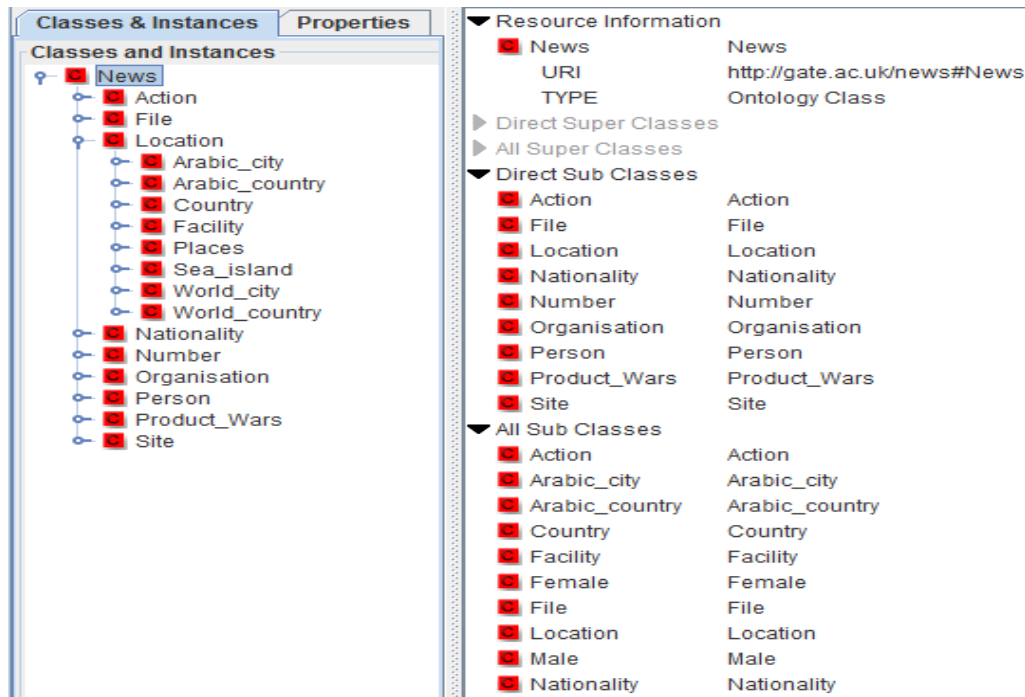


Figure (6.6): Classes and subclasses in news ontology

After visualizing the constructed ontology by VRs in GATE, we build a sample RDF store to represent information about resources on the text based on the ontology. We present base elements of the RDF model in the triple: a subject linked through a predicate to object. In RDF triple (S,P,O) We will say that <subject> has a property <predicate> valued. Part of the RDF is shown in Figure (6.8) where all taxonomic relations instance is transforms to object property such as "هي عاصمة", this property used to link between two classes ("القاهرة", "مصر").

```

1  @prefix : <http://gate.ac.uk/news#> .
2
3  @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
4  @prefix owl: <http://www.w3.org/2002/07/owl#> .
5  @prefix pext: <http://proton.semanticweb.org/protonext#> .
6  @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
7  @prefix psys: <http://proton.semanticweb.org/protonsys#> .
8  @prefix protons: <http://proton.semanticweb.org/2005/04/protons#> .
9  @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
10 @prefix protont: <http://proton.semanticweb.org/2005/04/protont#> .
11
12 <http://gate.ac.uk/classes#Country_هي_عاصمة_Country> a owl:ObjectProperty .
13
14 <http://gate.ac.uk/classes#القاهرة_هي_عاصمة_مصر> a owl:ObjectProperty .
15
16 مصر a <http://example.com/classes#Country> .
17
18 القاهرة a <http://example.com/classes#Country> .
19
20 <http://gate.ac.uk/classes#Places_جزء_من_Sea_island> a owl:ObjectProperty .
21
22 <http://gate.ac.uk/classes#البحر_الاحمر_جزء_من_المحيط_الهادي> a owl:ObjectProperty .
23

```

Figure (6.7): RDF triples as based on the ontology

6.7 Evaluation of the Approach

Automatic ontology building evaluation is a hard task because it is sometimes difficult to find triples for a given document or set of documents and usually depends on human experts. To ensure that the approach works well to represent ontologically Arabic Political News domain, we performed evaluation using two methods: first, using human expert review and second, using a reasoner in Protégé application. We used human expert review as reference to extract named entities and taxonomic relations from text to measure ontology correctness. After that, we get the results and calculate Precision (Eq. 3.2), Recall (Eq. 3.3), and F-measure (Eq. 3.4). Using a reasoner to check the consistency, i.e. to test whether or not one class is a subclass of another class.

6.7.1 Domain Expert Review VS the Proposed Approach

Domain experts assessed the correctness of the ontology in representing domain concepts and the relationships among them. We first evaluate the named entities and second we evaluate the taxonomic relations.

6.7.2 Named Entities Recognition and Human Evaluation

To evaluate Arabic named entity recognitions in GATE in Political News ontology domain, in one hand, human expert is used to extract terms from 50 Arabic news documents which were selected randomly from the main dataset that contained 750 news documents, and uploaded them in the real results corpus as described in Section (5.2.1). Figure (6.8) shows one document about Political News (as the target domain) selected for extracting NEs using a domain expert and perform evaluations for NEs extracted. The Table (6.2) shows an evaluation of this Arabic document with a comparison with the domain expert results and showing the R, P, F-measure manually. On the other hand, for the approach, we computed the three measurements Precision (P), Recall (R), and F-measure by Annotation Diff tool that found in GATE as shown in Figure (6.9).

الشرق الأوسط - وزراء الخارجية العرب يجتمعون في ليبيا تمهيدا للقمة العربية
 بدء اجتماعات وزراء الخارجية العرب في مدينة سرت تقع في ليبيا تمهيدا للقمة العربية الثانية والعشرين التي تنظم يومي السبت والاحد.
 وزراء الخارجية العرب يجتمعون في ليبيا تمهيدا للقمة العربية غادر وزير الخارجية العراقي هوشيار زيباري اجتماعا للجامعة العربية في طرابلس عاصمة ليبيا احتجاجا على اجراء القيادة الليبية
 في الليبي العقيد معمر القذافي قد استضاف في بداية الاسبوع مجموعة من المعارضين العراقيين بينهم مسؤولون اتصالات مع موالين للرئيس العراقي السابق صدام حسين. وكان الرئيس
 حزب البعث العراقي. وليس من الواضح فيما اذا كان زيباري سيغادر العاصمة الليبية عاندا الى العراق أم سينتظر انعقاد مؤتمر القمة العربي الذي سيبدأ جلساته السبت. وكانت قد بدأت في مدينة
 سرت الليبية اجتماعات وزراء الخارجية العرب تمهيدا للقمة العربية العادية الثانية والعشرين التي تنظم يومي السبت والاحد. يذكر انها المرة الاولى التي تعقد فيها القمة العربية في ضيافة الزعيم
 الليبي معمر القذافي الذي اشتهر باثارة الجدل في القمم العربية من خلال مواقف وتصريحات خارجة عن المألوف. ولكن ليبيا قد اعلنت مؤخرا رغبتها ب " وضع الخلافات العربية جانباً عنواناً للقمة
 الثانية والعشرين. وافتتح احمد بن عبدالله آل محمود وزير الدولة للشؤون الخارجية القطري أعمال الدورة 22 لمجلس اعمال وزراء الخارجية العرب. ودعا محمود الى ضرورة تحمل المسؤولية
 الكاملة للقيادة العرب تجاه ما يحدث المقدسات الاسلامية والمسيحية والعربية جزء من فلسطين والى ضرورة ان يتحمل المجتمع الدولي مسؤولياته تجاه ما يحدث في فلسطين من حصار جائر
 وتشريد وقتل تعتنت من قبل المحتل الاسرائيلي". مراجعة شاملة وأوضح محمود ان "العالم تعرض خلال الفترة الماضية الى التزامات عاتية ولان العالم العربي موجود في مهب هذه العواصف فان هذه
 الاضطرابات قد اثرت عليه بشكل مباشر وخلفت ازمة ثقة بين الاقطار العربية". كما طالب ب ضرورة مراجعة شاملة للاوضاع السياسية والاقتصادية العربية ولابد من وجود أجندة لهذه السياسات
 الامر الذي يتطلب عناية فائقة في معالجة كافة الاوضاع".
 سواء في السودان او الصومال والعراق وجزر القمر وما قدمته الدول المانحة استعرض الوزير القطري في كلمة الافتتاح ما قامت به الرئاسة للدورة المنصرمة خلال ترأسها للقمة العربية
 حسني مبارك بسبب المرض والرئيس اللبناني ميشال سليمان على الاقطار الى ضرورة التعاون الجاد. ومن أبرز الزعماء الغائبين عن القمة الرئيس المصري لجزر القمر، داعيا كافة
 خلفية المشاكل اللبنانية الليبية بسبب اختفاء الامام موسى الصدر عام 1978 بعدما شوهد للمرة الاخرة في العاصمة الليبية. كما أعلن مصدر رسمي اماراتي الخميس ان رئيس دولة الامارات وهو
 الشيخ خليفة بن زايد آل نهيان لن يشارك في القمة وسيستبدل عنه في رئاسة وفد بلاده حاكم ام القيوين الشيخ سعود بن راشد المعلا. ويشارك في الاجتماع وزراء خارجية الدول الاعضاء في
 الجامعة وعلى رأسهم وزير الدولة المستضيفة الليبي موسى كوسه، ومن أبرز الحاضرين وزير الخارجية السعودي وهو الامير سعود الفيصل و وزير الخارجية السوري وهو وليد المعلم. وقال
 مندوب سورية لدى الجامعة العربية يوسف احمد قبل دخوله الى الاجتماع ان "جميع المواضيع تقريبا قد تجزت من قبل المندوبين". و اضاف في تصريحات تناقشتها وكالات الانباء: "بقي مشروعان

Figure (6.8): Document from BBC News to Named Entity Recognition Evaluation

Table below show experimental result performed on the Political News document in Figure (6.8), the table consist of two column, the left column contain results of the domain expert, and the right column contain results of the proposed approach, the experimental result for one document that presented in Figure (6.8) is show under the table (6.1).

Table (6.1): Summary of Evaluation Based on the Domain Expert and the Proposed Approach for Extracting Named Entities

Results of the Domain Expert	Results of the Approach
الشرق الأوسط- وزراء الخارجية العرب- ليبيا- للقمة العربية- مدينة سرت- الثانية - وزير- الخارجية- الرئاسة- الجامعة- ام القيوين- العراقي- الجامعة العربية - طرابلس- الليبية- العقيد- معمر القذافي- استضاف- العراقي- صدام حسين- مجموعة- الطاهر - كوسه- العراقيين- حزب- العراق- مؤتمر- محمود- وزير- الدولة- الخارجية- المقدسات الاسلامية- المسيحية- فلسطين- المحتل- السودان- الصومال- الزعماء- المصري- حسني مبارك- اللبناني- ميشال سليمان- اللبنانية- الامام- موسى- اماراتي- الخميس- رئيس دولة- الامارات- الشيخ خليفة- آل نهيان- الشيخ- سعود- بن- راشد- الامير- سعود الفيصل- السوري- وليد- المعلم- مندوب- يوسف- القدس	الشرق الأوسط- وزراء الخارجية- العرب- ليبيا- العربية- مدينة سرت- الثانية - وزير الخارجية العراقي- الجامعة العربية - طرابلس- الليبية- العقيد- معمر القذافي- استضاف- العراقي- صدام- حسين- مجموعة- المعارضين- العراقيين- حزب- العراق- مؤتمر- محمود- وزير- الدولة- الخارجية- المقدسات الاسلامية- المسيحية- فلسطين- المحتل- السودان- الصومال- الزعماء- المصري- حسني مبارك- اللبناني- ميشال سليمان- اللبنانية- الامام- موسى- اماراتي- الخميس- رئيس دولة- الامارات- الشيخ خليفة- آل نهيان- الشيخ- سعود- بن- راشد- السعودي- الامير- سعود الفيصل- السوري- وليد- المعلم- مندوب- يوسف- القدس
Recall (R) = 91	Precision (P) = 87
F-measure (F) = 89	

Figure (6.9) shows the measurement of R, P and F using the Annotation diff tool in GATE for Table (6.2).

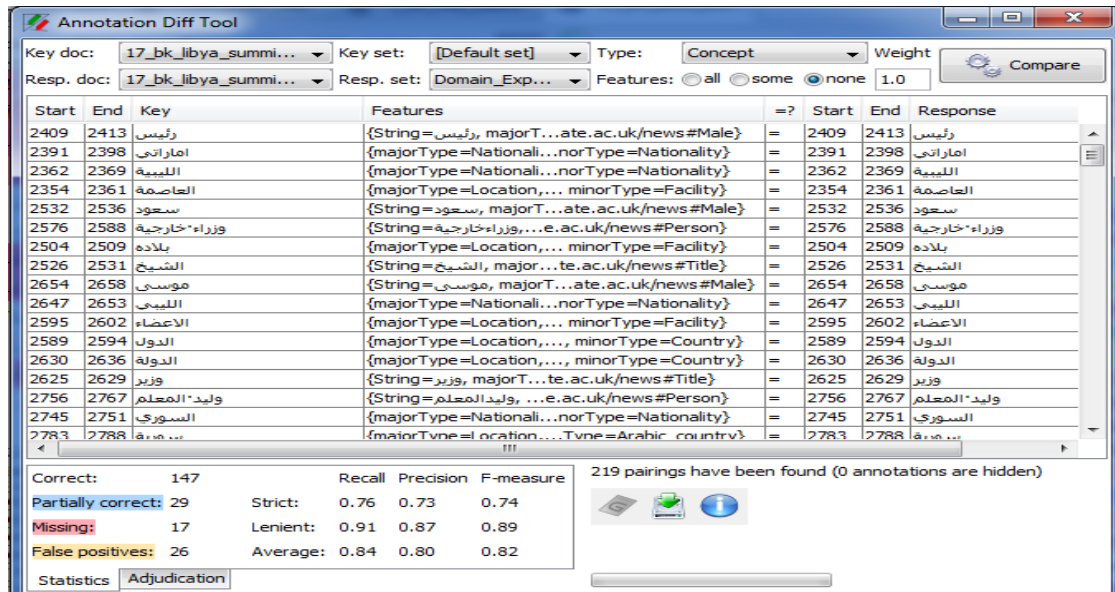


Figure (6.9): Named Entity Recognition Evaluation Using Annotation Diff

The results for all documents show that, Recall result is 93%, while the F-measure is 88% where recall refer to the number of correctly predicted items as a percentage of the total number of correct items for a given topic. The precision is 86% where precision refers to the number of correctly predicted items as a percentage of the number of items identified for a given topic. Precision is 86% because, usually the system have the ability to extract correct person name and locations from documents because most of person and location names in Gazetteer lists.

6.7.3 Taxonomic Relations Extraction and Human Evaluation

After evaluating named entities extraction as terms in news documents, we evaluate the effectiveness of the taxonomic relations extraction in the approach. We used domain expert to extract taxonomic relations between pairs of terms from the Political News corpus. After that, we applied our approach to the same documents and compared the results for each document with human results to compute the three measurements of P, R and F-measure for taxonomic relations extraction using the Annotation Diff tool as performed in the previous section. Finally, we compute the average results of R, P and F-measure for each.

The Figure (6.10) show one document about Political News selected to extract taxonomic relations between two pairs of NEs using domain expert. Table (6.3) shows the evaluation of Arabic Political News documents by comparing the approach results with domain expert results and compute R, P, F-measure to discover taxonomic relationships between terms. The measurement of R, P and F-measure for taxonomic relation evaluation is done using the Annotation diff in GATE as shown in Figure (6.11).

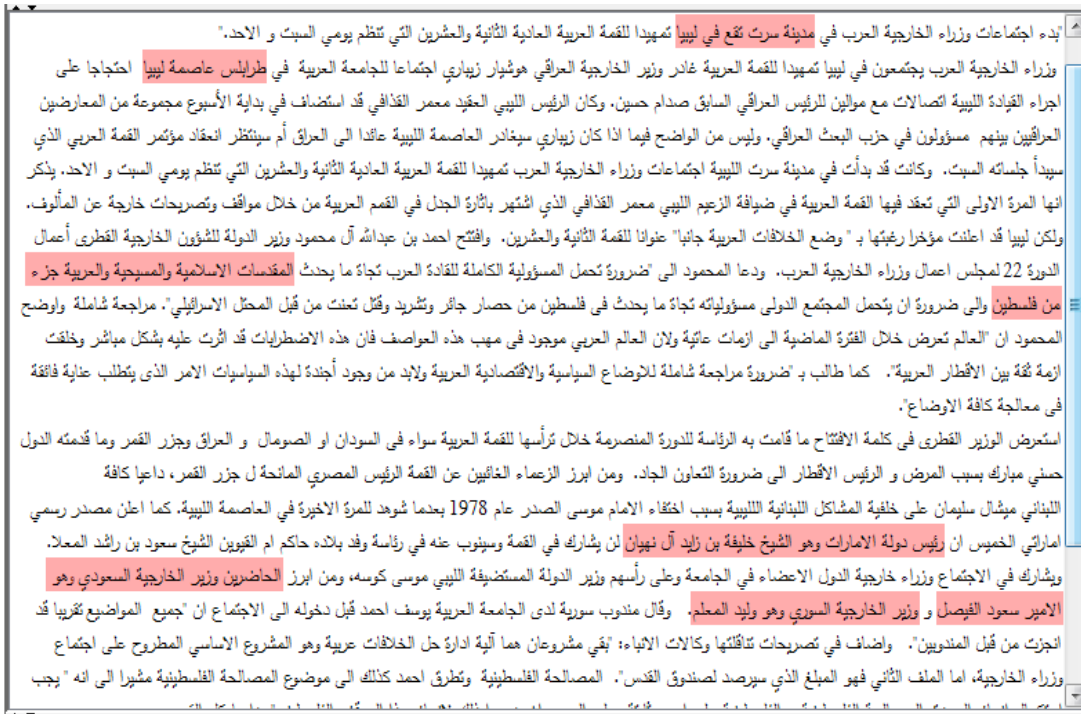


Figure (6.10): Document from BBC News to Taxonomic Relations Evaluation

Table below show experimental result performed on the Political News document in Figure (6.10) to extract taxonomic relations, the table consist of two column, the left column contain results of the domain expert for taxonomic relations extracted , and the right column contain results of the proposed approach for taxonomic relations extracted.

Table (6.2): Summary of Evaluation Based on the Domain Expert and the Proposed Approach for Extracting Taxonomic Relations

Results of the Domain Expert	Result of the Approach
مدينة سرت تقع في ليبيا وزير الخارجية العراقي هو هوشيار زيباري طرابلس عاصمة ليبيا الرئيس الليبي العقيد هو معمر القذافي المقدسات الاسلامية والمسحية والعربية جزء من فلسطين رئيس دولة الامارات وهو الشيخ خليفة بن زايد آل نهيان وزير الخارجية السعودي وهو الامير سعود الفيصل وزير الخارجية السوري وهو وليد المعلم	مدينة سرت تقع في ليبيا طرابلس عاصمة ليبيا المقدسات الاسلامية والمسحية والعربية جزء من فلسطين رئيس دولة الامارات وهو الشيخ خليفة بن زايد آل نهيان الحاضرين وزير الخارجية السعودي وهو الامير سعود الفيصل وزير الخارجية السوري وهو وليد المعلم
Recall (R) = 100	Precision (P) = 75
F-measure (F) = 86	

Figure (6.11) shows the measurement of R, P and F-measure for taxonomic relation using the Annotation diff tool in GATE shown in Table (6.3).

Start	End	Key	Features	=?	Start	End
2409	2460	رئيس دولة الامارات...و...يعة...بن زايد آل نهيان	{domain1=Person, rel...range_1_minor=Person}	=	2409	2460
1363	1414	المقدسات الاسلامية...و...العربية جزء من فلسطين	{relation=Part_Whole...omain_1_minor=Places}	=	1363	1414
376	394	طرابلس عاصمة ليبيا	{relation=Is_a, rela...inor=Arabic_country}	=	376	394
143	165	مدينة سرت تقع في ليبيا	{domain_1_Instance=مد...inor=Arabic_country}	=	143	165
2731	2767	وزير الخارجية السوري وهو وليد المعلم	{domain_1_Instance=وز...range_1_minor=Person}	=	2731	2767
2675	2728	الحاضرين وزير الخارج...و...الامير سعود الفيصل	{relation_String=وهو...main_1_minor=Surname}	~	2684	2728
				?	312	347
				?	494	521

Correct:	5	Recall	Precision	F-measure	8 pairings have been found (0 annotations are hidden)	
Partially correct:	1	Strict:	0.83	0.62		0.71
Missing:	0	Lenient:	1.00	0.75		0.86
False positives:	2	Average:	0.92	0.69		0.79

Figure (6.11): Taxonomic relations evaluation using Annotation Diff

The results show that, the average F-measure is 86%, Precision is 75% where precision refers to the number of correctly predicted items as a percentage of the number of items identified for a given topic. Recall result is 100%, where recall refer to the number of correctly predicted items as a percentage of the total number of correct items for a given topic. Recall is 100%, Because the approach have the ability to extract correct taxonomic relations from documents and depending on rules to extract all taxonomic relations. The source of errors in extracting the taxonomic relations because approach are not able to cover all taxonomic relations to express the relations, and the concepts statements are not arrangement in unified structure to extract triple statements.

Table (6.4) shows the results for all cases of documents have been computed in Corpus. The average of F-measure for all the chosen cases are considered the approach performance in ability to discover taxonomic relationships between terms.

Table (6.3): Summary the Results of Calculation R, P and F-measure for Extracting Taxonomic Relations

Annotation	Match	Only A	Only B	Overlap	Prec.B/A	Rec.B/A	F1.0-l.
Cdomain_Rel_Crange	93	12	11	39	0.9231	0.9167	0.9199

Figure (6.12) shows details for measures of R, P and F-measure in all documents for taxonomic relations extraction using Corpus Quality Assurance.

Document	Match	Only A	Only B	Overlap	Prec.B/A	Rec.B/A	F1.0-l.
32_lebanonnasrallah_tc2.txt_00030	4	0	0	0	1.0000	1.0000	1.0000
33_bk_lebanon_crash_ethiopian_airlines.txt_00031	2	1	1	1	0.7500	0.7500	0.7500
34_om_uae_lebanon_tc2.txt_00032	3	0	0	1	1.0000	1.0000	1.0000
35_yemen_houthsi_saudi_tc2.txt_00033	2	0	0	0	1.0000	1.0000	1.0000
36_af_britain_israel_tc2.txt_00034	5	0	0	0	1.0000	1.0000	1.0000
37_iraq_ramadi_explosion_tc2.txt_00035	1	0	0	1	1.0000	1.0000	1.0000
38_az_us_aid_palestinians_200m_tc2.txt_00036	3	0	0	1	1.0000	1.0000	1.0000
39_me_jerusalem_tc2.txt_00037	2	0	1	0	0.6667	1.0000	0.8000
3_somalia_mh_heavyfight.txt_00038	4	0	1	1	0.8333	1.0000	0.9091
40_om_brown_gaddafi_tc2.txt_00039	2	0	0	1	1.0000	1.0000	1.0000
41_hh_yemen_houthes_tc2.txt_0003A	3	0	0	0	1.0000	1.0000	1.0000
42_om_jordan_police_death_tc2.txt_0003B	1	0	0	1	1.0000	1.0000	1.0000
43_ah_jerusalem_settlement_tc2.txt_0003C	4	0	0	1	1.0000	1.0000	1.0000
44_mek_morocco_forum_livni_tc2.txt_0003D	2	0	0	0	1.0000	1.0000	1.0000
45_aq_syriaisrael_tc2.txt_0003E	1	0	1	0	0.5000	1.0000	0.6667
46_mr_hamas_delay_tc.txt_0003F	4	0	1	0	0.8000	1.0000	0.8889
47_mh_brown_sarkozi_gazareport_tc2.txt_00040	1	0	0	3	1.0000	1.0000	1.0000
48_bk_iraq_bagdad_blast.txt_00041	1	0	0	1	1.0000	1.0000	1.0000
49_aq_iraqhealthministry_tc.txt_00042	2	0	0	2	1.0000	1.0000	1.0000
4_ra_somalia_fighting_tc2.txt_00043	0	0	0	0	1.0000	1.0000	1.0000
50_am_uk_unions_israel_boycott_tc2.txt_00044	2	0	0	0	1.0000	1.0000	1.0000
5_ra_iran_shahroudi_tc2.txt_00045	3	0	1	1	0.8000	1.0000	0.8889
6_mek_iran_us_biden_tc2.txt_00046	1	0	0	1	1.0000	1.0000	1.0000
7_hh_israel_westbank_tc2.txt_00047	3	1	2	2	0.7143	0.8333	0.7692
8_as_iran_nuclear_tc2.txt_00048	0	0	0	1	1.0000	1.0000	1.0000
9_bk_assad_jumblat_lebanonf2b6.txt_00049	1	0	2	2	0.6000	1.0000	0.7500
R.txt_0004B	0	6	0	0	1.0000	0.0000	0.0000
Macro summary					0.9477	0.9513	0.9323
Micro summary	93	12	11	39	0.9231	0.9167	0.9199

Figure (6.12): Taxonomic Relations Evaluation Using Corpus Quality Assurance in GATE

6.7.4 Using Reasoner

One of the main services offered by a reasoner is to test whether or not one class is a subclass of another class. By performing such tests on the classes in an ontology it is possible for a reasoner to compute the inferred ontology class hierarchy. Another standard service that is offered by reasoners is consistency checking. The reasoner can check whether or not it is possible for the class to have any instances. A class is deemed to be inconsistent if it cannot possibly have any instances. Protégé offer OWL reasoners by plugged in. In order to demonstrate the use of the reasoner in detecting inconsistencies in the ontology we will open our ontology from Protégé application and using Reasoner to check ontology. Figure (6.13) show the ontology asserted hierarchies without any inconsistencies.

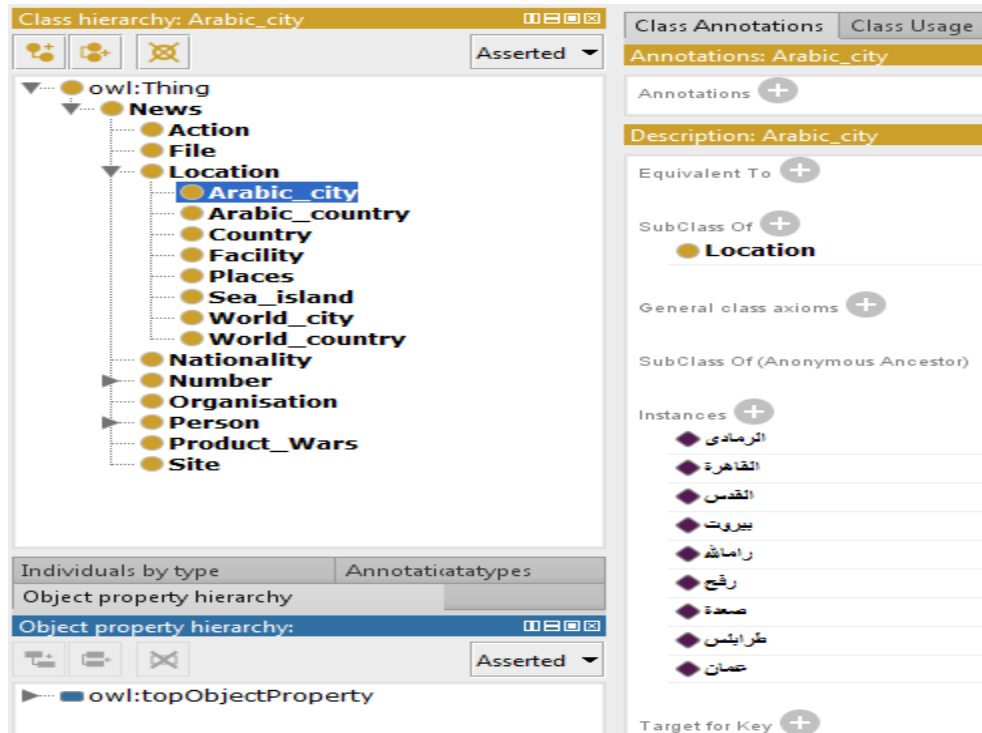


Figure (6.13): Consistency Ontology

Figure (6.14) shows the consistency for the ontological properties of taxonomic relations with pairs of concepts as domain and range.

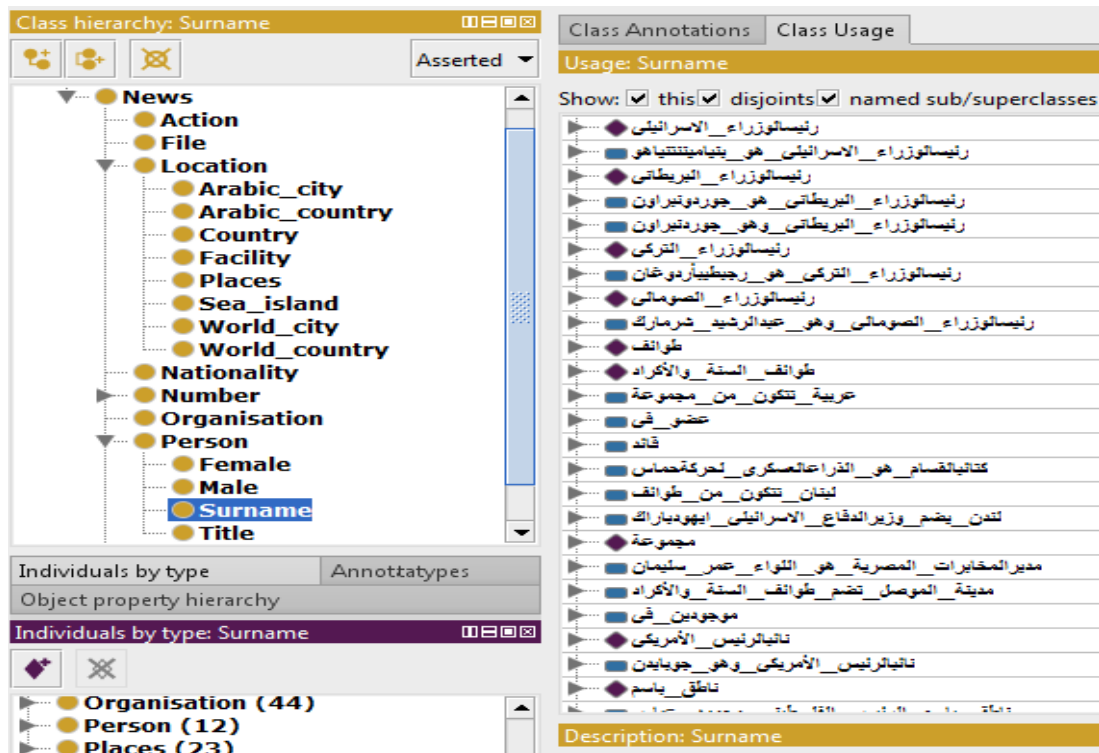


Figure (6.14): Consistency for the Properties of Taxonomic Relations

Figure (6.15) shows the OWLViz display of the asserted hierarchies of the resulting ontology structure for Political News domain.

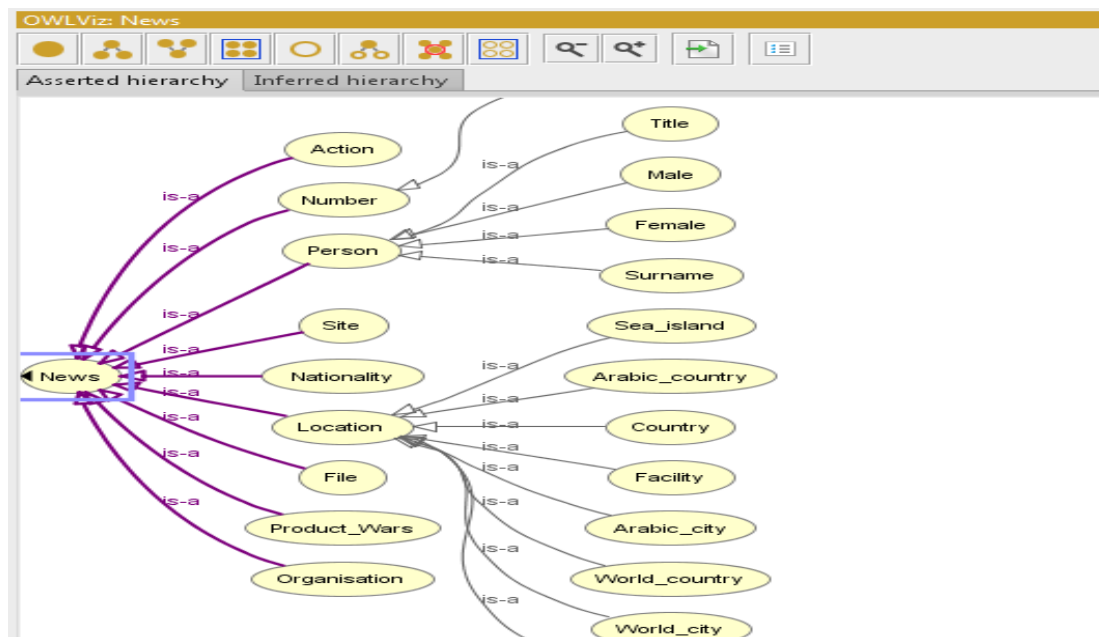


Figure (6.15): OWLViz Displaying the Asserted Hierarchy for the Ontology

6.8 Discussion

The results shown in Table 6.3, show number of taxonomic relations matching are 93 relations, number of taxonomic relations not matching are 11 relations, number of overlap relations are 39 relations for different annotation types. This is due to the following reasons:

- Taxonomic relations matching because the approach have the ability to extract correct taxonomic relations from documents depending on rules in specific domain to extract all taxonomic relations.
- Taxonomic relations not matching because approach are not able to cover all taxonomic relations patterns to express the relations, the concepts statements are not arrangement in unified structure to extract triple statements, and insufficiencies statements that contain (Subject, Predicate , Object).
- From all our experiments, we can say our approach achieved the best results for extracting taxonomic relations we indicated and shown in Table 6.3.

6.9 Summary

This chapter presented and analysed the experimental results. It stated the experimental setup and the experimental corpus characteristics. It also presented the text pre-processing results, and the results of the execution of the NER and the taxonomic relations extraction using GATE tool. After that, we illustrate the resulting ontology in the ontology visualizer and in OWL language representation. Finally, we presented the experimental results of constructing the Political News ontology and its evaluation based on Precision, Recall, and F-measure.

Chapter 7

Conclusions and Future Work

Chapter 7

Conclusions and Future Work

6.1 Summary

Exploiting knowledge present in textual documents is an important issue in building systems for knowledge management and ontology building. In this research, shown an approach for the automatic construction of ontology from a corpus of Political News domain for Arabic language. Information extraction techniques was used for recognized persons, locations, organizations and other name entities for extracting terms that denote elements of the ontology (concept, relation). Rules and patterns were used to extract taxonomic relations that bind two name entities as domain and range. The approach consists of pre-processing stage that includes encoding, tokenization, normalization, stop word removing, sentence splitting and then add some features such as POS and morphological analysing (light stemming). After that extract terms using simple integration between lexical resources and machine-learning classifier for Arabic named entity recognition. Taxonomic relations was identified by capturing some patterns with specific lexical elements in the text. Finally, construct a set of transformation rules, which were used to identify an appropriate ontological elements from the pattern extraction.

As already shown through this thesis, the proposed system has shown that the construction of ontology has the ability to achieve the following tasks:

- Extract named entities from Arabic Political News documents.
- The system is able to discover taxonomic relationships between named entities.
- Transform annotated concepts and relations in the text and create ontological concepts and resources.
- Build RDF store to represent information about resources on the text, and present the results as graph visualization which is considered useful for allowing the results of the relations between terms to be more readable.

For evaluation purposes, the three common effective measures were used; Recall, Precision and F-measure. The results of annotation achieves satisfactory results for all terms and taxonomic relation extractions. Precision is 86% and Recall is 93% for extracting named entities, Precision is 92% and Recall is 91% for

taxonomic relations extraction. Using the approach overcomes the problem of the manual construct ontology from Arabic documents. This means saving time and overcome difficulties with the manual process.

6.2 Contribution

The contributions of this research include:

- Automatic ontology construction based on document annotation for Arabic text. This includes building and evaluating a domain specific ontology namely " اخبار سياسية" (Political News).
- The approach for automatic ontology construction is based on Arabic document annotation. It helps users, in short time with high performance, to build domain ontology.
- The constructed ontology consists of taxonomic relations, particularly, class-subclass relationships and property-subproperty relationships.
- Based on the constructed taxonomic hierarchy, we built a sample RDF store consisting of triples related to the classes and properties of the constructed ontology.
- Adaptation of GATE to work with Arabic documents especially using machine learning for named entity recognition and building ontology.

6.3 Future Work

Although we achieved the objectives of our research, the results reveals the need for future work in the following directions:

- Extending the rules that are used in extracting taxonomic relations to be used as a basis for more specific relations and properties at the level of OWL such as symmetric/ asymmetric, cardinality, disjointness to name a few.
- Extracting other semantic relations such as non-taxonomic relations such as verb-based relations.
- Enriching the ontology construction using term synonyms based on linguistic resources.

References

- AbdelRahman, S., Elarnaoty, M., Magdy, M., & Fahmy, A. (2010). Integrated Machine Learning Techniques for Arabic Named Entity Recognition. *IJCSI*, 7, 27-36.
- Ahmed, Z. (2009). *Domain Specific Information Extraction for Semantic Annotation*. (Unpublished Master Thesis), Charles University.
- Al-Rajebah, N. I., & Al-Khalifa, H. S. (2014). Extracting Ontologies from Arabic Wikipedia: a linguistic approach. *Arabian journal for Science and Engineering*, 39(4), 2749-2771.
- Al-Thubaity, A. M., Khan, M., Alotaibi, S., & Alonazi, B. (2014). *Automatic Arabic Term Extraction from Special Domain Corpora*. Paper presented at International Conference IEEE on Asian Language Processing (IALP).
- Al Arfaj, A., & Al Salman, A. (2014). *Towards Ontology Construction from Arabic Texts-A Proposed Framework*. Paper presented IEEE at International Conference on Computer and Information Technology (CIT).
- Al Zamil, M. G., & Al-Radaideh, Q. (2014). Automatic Extraction Of Ontological Relations From Arabic Text. *Journal of King Saud University-Computer and Information Sciences*, 26(4), 462-472.
- Albukhitan, S., & Helmy, T. (2013). Automatic Ontology-based Annotation of Food, Nutrition and Health Arabic Web Content. *Procedia Computer Science*, 19, 461-469.
- Alias-i. (2008). *LingPipe 3.9.3*. Retrieved January 15, 2016, from: <http://alias-i.com/lingpipe>
- Almusaddar, M. Y. (2014). *Improving Arabic Light Stemming in Information Retrieval Systems*. (Unpublished Master Thesis), Islamic University, Gaza, Palestine.
- ANERGazet. (2008). Retrieved December 30, 2015, from: <http://users.dsic.upv.es/grupos/nle/?file=kop4.php>
- Arabic Stop Words. (2013). Retrieved March 6, 2016, from: <http://arabicstopwords.sourceforge.net/>
- Asharef, M., Omar, N., Albared, M., Minhui, Z., Weiming, W., Jingjing, Z., . . . Yu, F. (2012). Arabic Named Entity Recognition In Crime Documents. *Journal of Theoretical and Applied Information Technology*, 44(1), 1-6.
- Benajiba, Y., Diab, M., & Rosso, P. (2008). *Arabic Named Entity Recognition: An SVM-Based Approach*. Paper presented at the Proceedings of 2008 Arab International Conference on Information Technology (ACIT), Morocco.

- Blomqvist, E. (2005). Fully Automatic Construction of Enterprise Ontologies using Design Patterns: Initial method and first experiences On the Move to Meaningful Internet Systems 2005: *CoopIS, DOA, and ODBASE* (pp. 1314-1329). Italy: Springer.
- Bounhas, I., & Slimani, Y. (2009). *A Hybrid Approach for Arabic Multi-Word Term Extraction*. Paper presented at International Conference on Natural Language Processing and Knowledge Engineering, NLP-KE.
- Bozsak, E., Ehrig, M., Handschuh, S., Hotho, A., Maedche, A., Motik, B., . . . Stojanovic, L. (2002). KAON—Towards A Large Scale Semantic Web, *E-Commerce and Web Technologies* (pp. 304-313), Berlin Heidelberg : Springer.
- Buitelaar, P., Cimiano, P., & Magnini, B. (2005). *Ontology Learning from Text: methods, evaluation and applications* (Vol. 123): IOS press.
- Champin, P.-A. (2001). RDF Tutorial. *Pierre-Antoine Champin April 5*, 1-9.
- Chinchor, N., & Robinson, P. (1997). *MUC-7 Named Entity Task Definition*. Paper presented at the Proceedings of the 7th Conference on Message Understanding.
- Cimiano, P., & Völker, J. (2005). *Text2Onto Natural Language Processing and Information Systems* (pp. 227-238), Berlin Heidelberg : Springer.
- Corcho, O., Fernández-López, M., & Gómez-Pérez, A. (2003). Methodologies, Tools and Languages For Building Ontologies. Where is their meeting point?. *Data & knowledge engineering*, 46(1), 41-64.
- Correia, J., Girardi, R., & de Faria, C. G. (2011). *Extracting Ontology Hierarchies from Text (S)*. Paper presented at International Conferene on Software Engineering & Knowledge Engineering (SEKE).
- Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Ursu, C., Dimitrov, M., . . . Li, Y. (2009). *Developing Language Processing Components with GATE Version 5:(a User Guide)*: University of Sheffield.
- De Azevedo, R. R., Freitas, F., Rocha, R., Alves, J. A., Mário, C., & Rodrigues, O. Interactive Learning: an Approach for Building DL Ontologies. *Natural Language and Reasoning*, 1-6.
- El-Khair, I. A. (2006). Effects of Stop Words Elimination for Arabic Information Retrieval: a comparative study. *International Journal of Computing & Information Sciences*, 4(3), 119-133.
- Fader, A., Soderland, S., & Etzioni, O. (2011). *Identifying Relations For Open Information Extraction*. Paper presented at the Proceedings of the Conference on Empirical Methods in Natural Language Processing.
- Fouzi Harrag, Abdulwahab Alothaim, Abdulaziz Abanmy, Faisal Alomaigan, & Alsalehi, S. (2013). Ontology Extraction Approach for Prophetic Narration using

Association Rules. using Association Rules. *International Journal On Islamic Applications In Computer Science And Technology*, 1(2), 48-57.

Frank Manola, E. M. (2004). *RDF Primer*. Retrieved January 16, 2015, from <https://www.w3.org/TR/2004/REC-rdf-primer-20040210/>.

Gantayat, N. (2011). *Automated Construction of Domain Ontologies from Lecture Notes*. (Unpublished Master Thesis), Indian Institute of Technology, Bombay, Indian, Bombay.

Gruber, T. R. (1993). A Translation Approach To Portable Ontology Specifications. *Knowledge acquisition*, 5(2), 199-220.

Grycner, A., & Weikum, G. (2014). *HARPY: Hypernyms and Alignment of Relational Paraphrases*. Paper presented at the 25th International Conference on Computational Linguistics.

Hassanzadeh, K. (2013). *Converting Textual Documents to RDF Triples, Covering Syntactic and Semantic Structures*. (Unpublished Master Thesis), Alberta, Edmonton, Alberta.

Hazman, M., El-Beltagy, S. R., & Rafea, A. (2009). Ontology Learning from Domain Specific Web Documents. *International Journal of Metadata, Semantics and Ontologies*, 4(1-2), 24-33.

Hearst, M. A. (1992). *Automatic Acquisition of Hyponyms from Large Text Corpora*. Paper presented at the Proceedings of the 14th conference on Computational linguistics-Volume 2.

Kapociute-Dzikiene, J., Nøklestad, A., Johannessen, J. B., & Krupavicius, A. (2013, May 22-24). *Exploring Features for Named Entity Recognition in Lithuanian Text Corpus*. In Proceedings of the 19th Nordic Conference of Computational Linguistics.

Kiryakov, A., Popov, B., Ognyanoff, D., Manov, D., Kirilov, A., & Goranov, M. (2003, October). *Semantic Annotation, Indexing, and Retrieval*. Paper presented at the International Semantic Web Conference, Berlin Heidelberg : pringer.

Kristina Toutanova, D. K., Christopher Manning, and Yoram Singer. (2003). *Stanford Log-linear Part-Of-Speech Tagger*. Retrieved March 10, 2016, from <http://nlp.stanford.edu/software/tagger.shtml>

Lahbib, W., Bounhas, I., Elayeb, B., Evrard, F., & Slimani, Y. (2013). A hybrid Approach for Arabic Semantic Relation Extraction. Paper presented at Twenty-Sixth Florida Artificial Intelligence Research Society (FLAIRS) Conference, Florida.

Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., . . . Heitz, T. (2001). *Developing Language Processing Components with GATE*

Version 7 (a User Guide). University of Sheffield, UK, Web: <http://gate.ac.uk/sale/tao/index.html>.

- Mazari, A. C., Aliane, H., & Alimazighi, Z. (2012). *Automatic Construction of Ontology from Arabic Texts*. Paper presented at the ICWIT, Malaysia.
- McGuinness, D. L., & Van Harmelen, F. (2004). OWL Web Ontology Language Overview. *W3C recommendation*, 10(10), 1-6.
- Mezghanni, I. B., & Gargouri, F. (2014). *Learning of Legal Ontology Supporting the User Queries Satisfaction*. Paper presented at the Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 01.
- Nadeau, D., & Sekine, S. (2007). A Survey of Named Entity Recognition and Classification. *Linguisticae Investigationes*, 30(1), 3-26.
- Nakashole, N., Weikum, G., & Suchanek, F. (2012). *PATTY: A Taxonomy of Relational Patterns with Semantic Types*. Paper presented at the Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning.
- Nguyen, D. Q., Nguyen, D. Q., Ma, K. T., & Pham, S. B. (2011). *Automatic Ontology Construction from Vietnamese Text*. Paper presented at the Natural Language Processing and Knowledge Engineering (NLP-KE), 2011 7th International Conference on.
- Oudah, M., & Shaalan, K. F. (2012). *A Pipeline Arabic Named Entity Recognition using a Hybrid Approach*. Paper presented at International Conference COLING on Computational Linguistics, Ireland.
- Pandit, S. (2010). Ontology-Guided Extraction of Structured Information from Unstructured Text: *Identifying and capturing complex relationships*.
- Ponzetto, S. P., & Strube, M. (2007). *Deriving a Large Scale Taxonomy from Wikipedia*. Paper presented at the AAAI, British .
- Porter, M. F. (1980). An Algorithm for Suffix Stripping. *Program*, 14(3), 130-137.
- Ribeiro de Azevedo, R., Freitas, F., Rocha, R. G., Alves de Menezes, J. A., de Oliveira Rodrigues, C. M., & Silva, G. D. F. (2014). *An Approach for Learning and Construction of Expressive Ontology from Text in Natural Language*. Paper presented at the Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on.
- Ryu, P.-M., & Choi, K.-S. (2006). *Taxonomy Learning using Term Specificity and Similarity*. Paper presented at the Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge.

- Saad, M. K., & Ashour, W. (2010). Arabic Morphological Tools for Text Mining. *Corpora*, 18, 1-19.
- Sean Bechhofer, F. v. H., Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, Lynn Andrea Stein. (2009). *OWL Web Ontology Language*. Retrieved November, 2015, from <https://www.w3.org/TR/2004/REC-owl-ref-20040210/>
- Shalan, K. (2010). Rule-Based Approach in Arabic Natural Language Processing. *The International Journal on Information and Communication Technologies (IJICT)*, 3(3), 11-19.
- Shalan, K. (2014). A Survey of Arabic Named Entity Recognition and Classification. *Computational Linguistics*, 40(2), 469-510.
- Shamsfard, M., & Barforoush, A. A. (2003). The State of the Art in Ontology Learning: a framework for comparison. *The Knowledge Engineering Review*, 18(04), 293-316.
- Thakker, D., Osman, T., & Lakin, P. (2009). Gate Jape Grammar Tutorial. *Nottingham Trent University, UK, Phil Lakin, UK, Version, 1*.
- Unicode 8.0 Character Code Charts. (2015, July). Retrieved March 3, 2016, from <http://www.unicode.org/charts/>
- Unicode Windows 1256. (2015, July). Retrieved March 1, 2016, from <https://msdn.microsoft.com/en-us/goglobal/cc305149>
- Wang, T., Li, Y., Bontcheva, K., Cunningham, H., & Wang, J. (2006). *Automatic Extraction of Hierarchical Relations from Text*. Paper Presented on European Semantic Web Conference (pp. 215-229). Berlin Heidelberg : Springer.
- wikipedia. (2015, August). *Universal Quantification*. Retrieved April 20, 2016, from https://en.wikipedia.org/wiki/Universal_quantification
- Zaidi, S., Laskri, M., & Abdelali, A. (2010). *Arabic Collocations Extraction Using Gate*. Paper presented at International Conference IEEE on Machine and Web Intelligence (ICMWI), 2010.
- Zayaraz, G. (2015). Concept Relation Extraction Using Naïve Bayes Classifier for Ontology-Based Question Answering Systems. *Journal of King Saud University-Computer and Information Sciences*, 27(1), 13-24.

Appendix: JAPE Rules for Ontology Construction

A. Main JAPE Rule to Show all Phases

```
1  /*
2  * Main.jape
3  *
4  */
5  multiphase: MainGrammar
6  Phases:
7  Concepts
8  Taxonomic_Relation
9  Taxonomic_Relation_Feature
10 /*
11  Domain_Rel_Range
12 */
13 Cdomain_Rel_Crange
14 Transform
15 /*
16 TransformClass
17 TransformInstance*/
```

B. Location Concept Extraction using JAPE Rules.

Phase: Concepts

Input: Lookup Token

Options: control = appelt

```
/* First Concept For Location */
/*****
```

Rule: Location

```
(
{Lookup.majorType == "location"} |
{Lookup.majorType == "Facility"} |
{Lookup.majorType == "facility"} |
{Lookup.majorType == "Gpe"}
):mention
-->
:mention{
Annotation mentionAnn = mentionAnnots.iterator().next();
String cString="";
// get String
Long cStart = mentionAnn.getStartNode().getOffset();
Long cEnd = mentionAnn.getEndNode().getOffset();
```

```

AnnotationSet toks = inputAS.get("Token", cStart, cEnd);
List<Annotation> orderedToks = gate.Utils.inDocumentOrder(toks);
    for(Annotation a : orderedToks){
        cString = cString + a.getFeatures().get("string");
    }
//find the class of the mention
String majorType =(String)
mentionAnn.getFeatures().get(gate.creole.ANNIEConstants.LOOKUP_MAJOR_TY
PE_FEATURE_NAME);
majorType = majorType.substring(0, 1).toUpperCase()+majorType.substring(1) ;
String className =(String)
mentionAnn.getFeatures().get(gate.creole.ANNIEConstants.LOOKUP_CLASS_FE
ATURE_NAME);
String minorType =(String)
mentionAnn.getFeatures().get(gate.creole.ANNIEConstants.LOOKUP_MINOR_TY
PE_FEATURE_NAME);

// create the ontology and class features
    String locType = (String)mentionAnn.getFeatures().get(majorType);
    if(locType == null) locType = "location";
        String Oclass = ontology.getDefaultNameSpace()+locType.substring(0,
1).toUpperCase()+locType.substring(1) ;
String Mclass = "";

FeatureMap features = Factory.newFeatureMap();
features.put("majorType", "Location" );
features.put("String", cString );
features.put("classes", "مكان" );
features.put("class", Oclass);
if(minorType.length() != 0)
{ minorType = minorType.substring(0, 1).toUpperCase()+minorType.substring(1);
    String locTypeM = minorType;
Mclass=ontology.getDefaultNameSpace()+locTypeM.substring(0,1).toUpperCase()+
locTyM.substring(1);
features.put("classM", Mclass);
features.put("minorType", minorType);
}
// create the new annotation
try {
outputAS.add(mentionAnnots.firstNode().getOffset(),
mentionAnnots.lastNode().getOffset(), "Concept", features);
}
catch(InvalidOffsetException e) {
throw new JapeException(e);}

```

C. Taxonomic Relations Extraction using JAPE Rules.

phase: Taxonomic_Relation

Input: Token

options: control = appelt

Rule: Is_a

```
(
{Token.string == "هو" }({Token.string == "احد" })? |
{Token.string == "هي" }({Token.string == "احدى" })? |
{Token.string == "مقبلة" }({Token.string == "على" })? |
{Token.string == "هي" }({Token.string == "عاصمة" })? |
{Token.string == "هو" } | {Token.string == "وهو" } | {Token.string == "هي" } |
{Token.string == "هم" } | {Token.string == "هما" } |
{Token.string == "هم" } {Token.string == "من" } |
{Token.string == "هؤلاء" } | {Token.string == "هن" } |
{Token.string == "عاصمة" }
):mention
-->
:mention{
gate.AnnotationSet predi = (gate.AnnotationSet) bindings.get("mention");
gate.FeatureMap features = Factory.newFeatureMap();
features.put("rule", "Is_a");
outputAS.add(predi.firstNode(), predi.lastNode(), "Taxonomic_Relation", features);
}
//-----
```

Rule: Cause_Effect

```
(
{Token.string == "بسبب" } | {Token.string == "يسبب" } | {Token.string == "نتيجة" } |
{Token.string == "سبب" }
):cause
-->
{
gate.AnnotationSet cause = (gate.AnnotationSet) bindings.get("cause");
gate.FeatureMap features = Factory.newFeatureMap();
features.put("rule", "Cause_Effect");
outputAS.add(cause.firstNode(), cause.lastNode(), "Taxonomic_Relation", features);
}
//-----
```

Rule: Part_Whale

```
(
{Token.string == "عضو" } {Token.string == "في" } | {Token.string == "تتكون" }
{Token.string == "من" } | {Token.string == "يتكون" } {Token.string == "من" } |
{Token.string == "جزء" } {Token.string == "من" } | {Token.string == "تحتوي" }
{Token.string == "علي" } |
{Token.string == "تشمل" } {Token.string == "على" } |
{Token.string == "احد" } {Token.string == "اعضاء" } | {Token.string == "احد" }
{Token.string == "احزاب" } |

```



```

{Token.string == "من" }{Token.string == "فصائل" } |
({Token.string == "تنقسم" }|{Token.string == "ينقسم" }){Token.string == "الى" } |
({Token.string == "تتألف" }|{Token.string == "يتألف" }){Token.string == "من" } |
({Token.string == "تنتمي" }|{Token.string == "ينتمي" }){Token.string == "الى"
}|{Token.string == "الى" } |
{Token.string == "من" }{Token.string == "مكونات" } |
{Token.string == "من" }{Token.string == "فصيلة" }
):part
-->
{
gate.AnnotationSet part = (gate.AnnotationSet) bindings.get("part");
gate.FeatureMap features = Factory.newFeatureMap();
features.put("rule","Part_Whale");
outputAS.add(part.firstNode(),part.lastNode(),"Taxonomic_Relation",features);
}
//-----
Rule: Has_a
(
{Token.string == "له" } | {Token.string == "لها" } |
{Token.string == "التابعة" }|{Token.string == "تضم" }|{Token.string == "يضم" }|
{Token.string == "تقع" }{Token.string == "في" } |
{Token.string == "يقع" }|{Token.string == "تقع" } |
({Token.string == "موجود" }|{Token.string == "موجودة" }|{Token.string == "الموجودة"
})|{Token.string == "في" } |
({Token.string == "موجود" }|{Token.string == "موجوده" }|{Token.string == "الموجوده"
})|{Token.string == "موجودين" }){Token.string == "في" } |
{Token.string == "توجد" }{Token.string == "في" } |
({Token.Root== "ضم" }|{Token.Root=="حوى" } ) |
{Token.string==" طراز"}|({Token.string=="من"}{Token.string==" طراز"})
):has
-->
{
gate.AnnotationSet has = (gate.AnnotationSet) bindings.get("has");
gate.FeatureMap features = Factory.newFeatureMap();
features.put("rule","Has_a");
outputAS.add(has.firstNode(),has.lastNode(),"Taxonomic_Relation",features);
}
//-----
Rule: Kind_of
(
{Token.string == "نوع" }{Token.string == "من" } |
{Token.string == "مثل" }|
{Token.string == "احد" }{Token.string == "النواع" }
):kind
-->
{
gate.AnnotationSet kind = (gate.AnnotationSet) bindings.get("kind");
gate.FeatureMap features = Factory.newFeatureMap();

```

```

features.put("rule", "Kind_of");
outputAS.add(kind.firstChild(), kind.lastNode(), "Taxonomic_Relation", features);
}
//-----
Rule: Tital
(
{Token.string == "رئيس" } || {Token.string == "الرئيس" } || {Token.string == "هو"
}{Token.string == "رئيس" }|
{Token.string == "قائد" } || {Token.string == "القائد" }|
{Token.string == "امير" } | {Token.string == "الامير" }
):tital
-->
{
gate.AnnotationSet kind = (gate.AnnotationSet) bindings.get("tital");
gate.FeatureMap features = Factory.newFeatureMap();
features.put("rule", "Tital");
outputAS.add(kind.firstChild(), kind.lastNode(), "Taxonomic_Relation", features);
}
-----

```

D. Taxonomic Relations Features Extraction using JAPE Rules.

```

Phase: Taxonomic_Relation_Feature
Input: Taxonomic_Relation
Options: control = appelt
Rule: Taxonomic_Rel_Feature
(
{Taxonomic_Relation}
):mention
-->
:mention{
gate.AnnotationSet predi = (gate.AnnotationSet) bindings.get("mention");
gate.FeatureMap features = Factory.newFeatureMap();
Annotation mentionAnn = mentionAnnots.iterator().next();
// find the class of the mention
String rule_name = (String)mentionAnn.getFeatures().get("rule");
// find the text covered by the annotation
String mentionName;
try{ mentionName =
doc.getContent().getContent(mentionAnn.getStartNode().getOffset(),
mentionAnn.getEndNode().getOffset()).toString();
}
catch(InvalidOffsetException e){
throw new GateRuntimeException(e); //This should never happen
}
mentionName = mentionName.replace(" ", "_");
features.put("rule", rule_name);
}

```

```

features.put("string", mentionName);
outputAS.add(predi.firstNode(),predi.lastNode(),"Taxonomic_Relation_Feature",features);
}

```

E. Transformation JAPE Rules.

```

/*
 * Transform.jape
 */
Phase: Transform
Input: Cdomain_Rel_Crange //Domain_Rel_Range
Options: control = first //appelt
Rule: Transform
({Cdomain_Rel_Crange}):relationIden
-->
:relationIden{
//build the first Node in ontology .
OURI aURI1 = ontology.createOURI("http://gate.ac.uk/news#News");
OClass SuperClass = ontology.addOClass(aURI1);
Annotation theInstance = (Annotation)relationIdenAnnots.iterator().next();
//get the domain strings from the features of Annotaiton
String domain = theInstance.getFeatures().get("domain1").toString();
String domain_Instance =
theInstance.getFeatures().get("domain1_Instance").toString();
String domain_Minor = "";
if(theInstance.getFeatures().get("domain1_minor") != null){
domain_Minor = theInstance.getFeatures().get("domain1_minor").toString();

if(theInstance.getFeatures().get("domain2") != null){
domain = theInstance.getFeatures().get("domain2").toString();
domain_Instance =
domain_Instance+"_"+theInstance.getFeatures().get("domain2_Instance").toString();
if(theInstance.getFeatures().get("domain2_minor") != null){
domain_Minor = theInstance.getFeatures().get("domain2_minor").toString(); }
}
if(theInstance.getFeatures().get("domain3") != null){
domain = theInstance.getFeatures().get("domain3").toString();
domain_Instance =
domain_Instance+"_"+theInstance.getFeatures().get("domain3_Instance").toString();
if(theInstance.getFeatures().get("domain3_minor") != null){
domain_Minor = theInstance.getFeatures().get("domain3_minor").toString();}
}
// to improve the ontology
domain = theInstance.getFeatures().get("domain1").toString();
if(theInstance.getFeatures().get("domain1_minor") != null){
domain_Minor = theInstance.getFeatures().get("domain1_minor").toString(); }
}

```

```

//get the range strings from the features of Annotaiton
String range = theInstance.getFeatures().get("range1").toString();
String range_Instance = theInstance.getFeatures().get("range1_Instance").toString();
String range_Minor="" ;
if(theInstance.getFeatures().get("range1_minor") != null){
range_Minor = theInstance.getFeatures().get("range1_minor").toString();}
if(theInstance.getFeatures().get("range2") != null){
    range = theInstance.getFeatures().get("range2").toString();
    range_Instance =
range_Instance+"_"+theInstance.getFeatures().get("range2_Instance").toString();
if(theInstance.getFeatures().get("range2_minor") != null){
    range_Minor = theInstance.getFeatures().get("range2_minor").toString();}
}
if(theInstance.getFeatures().get("range3") != null){
    range = theInstance.getFeatures().get("range2").toString();
    range_Instance =
range_Instance+"_"+theInstance.getFeatures().get("range3_Instance").toString();
if(theInstance.getFeatures().get("range3_minor") != null){
    range_Minor = theInstance.getFeatures().get("range3_minor").toString();}
}

// to improve the ontology
range = theInstance.getFeatures().get("range1").toString();
if(theInstance.getFeatures().get("range1_minor") != null){
    range_Minor = theInstance.getFeatures().get("range1_minor").toString();}

//get the Relation strings from the features of Annotaiton
String Rel = theInstance.getFeatures().get("relation_String").toString();
String Rel3 = theInstance.getFeatures().get("relation_String").toString();

    domain = domain.replace(" ", "_") ;
    domain_Instance = domain_Instance.replace(" ", "_") ;
    domain_Minor = domain_Minor.replace(" ", "_") ;

    range = range.replace(" ", "_") ;
    range_Instance = range_Instance.replace(" ", "_") ;
    range_Minor = range_Minor.replace(" ", "_") ;

    String Oproperty = Rel; //domain +"_" + Rel +"_" + range;
    String Oproperty3 = domain_Instance +"_" + Rel +"_" + range_Instance;
    String Oproperty_Minor = domain_Minor +"_" + Rel +"_" + range_Minor;

// Create URI for domain and range.
gate.creole.ontology.OURI domclassURI =
ontology.createOURI("http://example.com/classes#" + domain);
gate.creole.ontology.OURI rngclassURI =
ontology.createOURI("http://example.com/classes#" + range);

```

```

gate.creole.ontology.OURI domMclassURI =
ontology.createOURI("http://example.com/classes#" + domain_Minor);
gate.creole.ontology.OURI rngMclassURI =
ontology.createOURI("http://example.com/classes#" + range_Minor);

//Add domain and range concept to ontology
gate.creole.ontology.OClass Domain = ontology.addOClass(domclassURI);
SuperClass.addSubClass(Domain);

gate.creole.ontology.OClass Range = ontology.addOClass(rngclassURI);
SuperClass.addSubClass(Range);

gate.creole.ontology.OClass DomainM = ontology.addOClass(domMclassURI);
Domain.addSubClass(DomainM);

gate.creole.ontology.OClass RangeM = ontology.addOClass(rngMclassURI);
Range.addSubClass(RangeM);

//check if property exist then
gate.creole.ontology.ObjectProperty OP =
ontology.getObjectProperty(ontology.createOURI("http://example.com/classes#" +
Oproperty));
gate.creole.ontology.ObjectProperty OP3 =
ontology.getObjectProperty(ontology.createOURI("http://example.com/classes#" +
Oproperty3));
gate.creole.ontology.ObjectProperty OP_M =
ontology.getObjectProperty(ontology.createOURI("http://example.com/classes#" +
Oproperty_Minor));

if(OP == null )
{
// Create Domain and Range Sets and add Domain and Range classes
Set<gate.creole.ontology.OClass> theDomain = new
HashSet<gate.creole.ontology.OClass>();
Set<gate.creole.ontology.OClass> theRange = new
HashSet<gate.creole.ontology.OClass>();
//// the class you have for the domain
theDomain.add(Domain);
theDomain.add(DomainM);
//// the class you have for the range
theRange.add(Range);
theRange.add(RangeM);

gate.creole.ontology.URI uri =
gate.creole.ontology.OntologyUtilities.createURI(ontology, domain_Instance, false);
if(!ontology.containsOInstance(uri)) {
//create the instance in the ontology
ontology.addOInstance(uri, DomainM); }

```

```

gate.creole.ontology.URI uri_range =
gate.creole.ontology.OntologyUtilities.createURI(ontology,range_Instance, false);
if(!ontology.containsOInstance(uri_range)) {
//create the instance in the ontology
ontology.addOInstance(uri_range,RangeM);}

// create the URI for the new property:
ontology.addObjectProperty(ontology.createOURI("http://gate.ac.uk/classes#" +
Oproperty), theDomain, theRange);

// create the URI for the property3:
ontology.addObjectProperty(ontology.createOURI("http://gate.ac.uk/classes#" +
Oproperty3), theDomain, theRange);

// create the URI for the property3:
ontology.addObjectProperty(ontology.createOURI("http://gate.ac.uk/classes#" +
Oproperty_Minor), theDomain, theRange);
}
else {
Set<gate.creole.ontology.OResource> theDomain= OP.getDomain();
theDomain.add(Domain);
theDomain.add(DomainM);
Set<gate.creole.ontology.OResource> theRange= OP.getRange();
theRange.add(Range);
theRange.add(RangeM);

// OP3
Set<gate.creole.ontology.OResource> theDomain3= OP3.getDomain();
theDomain3.add(Domain);
Set<gate.creole.ontology.OResource> theRange3= OP3.getRange();
theRange3.add(Range);

// OP_M
Set<gate.creole.ontology.OResource> theDomain_M= OP_M.getDomain();
theDomain_M.add(DomainM);
Set<gate.creole.ontology.OResource> theRange_M= OP_M.getRange();
theRange_M.add(RangeM);

System.err.println("object property has A exists"); }
}

```

F. Configuration file parameters for Machine Learning:

- URL of the configuration file: we specify the location of XML configuration file.
- Corpus: corpus contains the documents that the PR uses in the training process.
- InputASName: the annotation set containing the annotations for the linguistic features and the class labels.
- OutputASName: the resulting annotation set which are the results of applying the SVM ML on the InputASName annotation set.
- LearningMode: the set of the following values ("TRAINING", "APPLICATION", "EVALUATION").

Settings in the Batch Learning PR XML configuration file are the following:

- SURROUND: when named entity recognition span of several tokens is to be identified. `<SURROUND value="true"/>`.
- EVALUATION: when parameter learning mode is "EVALUATION", it will split the documents into two parts, the training dataset and the test dataset, to measure of success, the item method determines which method to use for evaluation. We select k-fold, in k-fold cross-validation the PR segments the corpus into k partitions of equal size, and uses each of the partitions in turn as a test set, with all the remaining documents as a training set.

`<EVALUATION method="kfold" runs="4"/>`

- multiClassification2Binary: many algorithms are binary classifiers (e.g. yes/no), but we have several classes (Person, Location, Organization etc.), therefore the problem must be converted to a set of binary problems, so we use binary algorithms one-vs.-others as (LOC vs. PERS+ORG / PERS vs. LOC+ORG / ORG vs. LOC+PERS).

`<multiClassification2Binary method="one-vs-others"/>`

- thresholdProbabilityBoundary: how likely a result is to be correct, is a threshold for the beginning and end instances.

`<PARAMETER name="thresholdProbabilityBoundary" value="0.4"/>`

- thresholdProbabilityEntity: how likely a result is to be correct, is a threshold for beginning and end instances combined

`<PARAMETER name="thresholdProbabilityEntity" value="0.2"/>`

- ENGINE: it specifies which machine learning algorithm we wish to use, we are using the SVM.

```
<ENGINE nickname="SVM" implementationName="SVMLibSvmJava"
options=" -c 0.7 -t 0 -m 100 -tau 0.5 "/>
```

Setting the DATASET Element that defines the type of annotation to be used as training instance and the set of attributes instances. In XML configuration file.

- INSTANCE-TYPE: we tell the ML PR what our instance annotation is, the goal is try to learn how the attributes of every instance relate to its class, Token annotation have all attribute for learning. This attribute (POS, Root , kind, string, length , next and previous token and POS).

```
<INSTANCE-TYPE>Token</INSTANCE-TYPE>
  <!-- Attribute Instance -->
  <ATTRIBUTELIST>
    <NAME>POS</NAME>
    <SEMTYPE>NOMINAL</SEMTYPE>
    <TYPE>Token</TYPE>
    <FEATURE>category</FEATURE>
    <RANGE from="-2" to="2"/>
  </ATTRIBUTELIST>
```

- ATTRIBUTE: to specify the class attribute, so we tells the Batch Learning that is the class attribute to learn.

```
<ATTRIBUTE> <NAME>Class</NAME>
  <SEMTYPE>NOMINAL</SEMTYPE>
  <TYPE>Target</TYPE>
  <FEATURE>type</FEATURE>
  <POSITION>0</POSITION>
</CLASS/> </ATTRIBUTE>
```

----- End of the Thesis -----