Deanship of Graduate Studies

Al-Quds University

Extracting Learning Objectives from Arabic Text Based on Bloom's Taxonomy

Tareq Issam Abdel-Azeez Ali

M.Sc. Thesis

Jerusalem-Palestine

1440-2019

Extracting Learning Objectives from Arabic Text Based on Bloom's Taxonomy

Prepared by:

Tareq Issam Abdel-Azeez Ali

B.Sc. Computer Science / Najah university /Palestine

Supervisor: Dr. Badie Sartawi

A thesis submitted in fulfillment of requirements for the degree of Master of Computer Science of Al-Quds University.

1440/2019

Al-Quds University

Deanship of Graduate Studies

Computer Science

Thesis Approval

Extracting Learning Objectives from Arabic Text Based on Bloom's Taxonomy

Student Name: Tareq Issam Abdel-Azeez Ali

Registration No: 21411042

Supervisor: Dr. Badie Sartawi

Master thesis submitted and accepted date: 23/6 /2019

The names and signatures of the examining committee members are as follows:

1-Head of committee: Dr. Badie Sartawi signature:..................

2- Internal Examiner: Dr. Rashid Jayousi signature:..................

3- External Examinar: Dr. Radwan Tahboub signature:..................

Jerusalem- Palestine

1440/2019

Dedication

This thesis is dedicated to my parents, my brothers and sisters and my wife who have always loved me unconditionally and who was good examples have taught me to work hard for the things that i aspire to achieve I am truly grateful for having in my life. This work is also dedicated to my teachers who have always been a constant source of support and encouragement during the challenges of my whole college life.

Declaration

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

Signed ………………..

Tareq Issam Abdel-Azeez Ali

Date: 23 / 6 / 2019

Acknowledgment

First of all, I have to thank Allah and praise him so much for his infinite graces, Allah gave me patience and assistance to accomplish this research, and Allah taught me what I did not know, so Alhamdulillah.

Many thanks and sincere gratefulness to the teachers Dr. BADIE SARTAWI for his efforts, guidance and assistance during the work in this thesis and before, and I would like to thank the rest of the thesis committee and the department of computer science for all their support during my research.

Special thanks also to my parents, brothers, sisters, my wife and my lovely little daughter 'Reem' for their continued love, trust and support, without them I would never have been able to achieve what I have achieved.

Abstract

Arabic language one of the most complex and comprehensive languages, and the need to find a uniform and accurate pattern for analysis and understanding the language has become an urgent necessity in light of this technological acceleration, which can help in several areas, including education and research, and this may be helpful in trying to automating the education using the methodology of learning objectives and learning outcomes. The aim of this thesis is to analyze Arabic text in order to extract the meaningful learning objectives from the text based on the first two level from bloom's Taxonomy, remembering and understanding, using two main methodologies the first one by building Arabic Ontology (a primitive tree in this stage), which include Arabic words classification a fixed and unified classification based on the origin of the word, the second methodology is by adoption a standard templates for the learning objectives each template include several keywords and indicators then using three stages. First stage perform pre-processing operations on the text such as divide the text into (paragraphs, sentences, words), remove/set diacritic marks, exclude some words before processing, the second stage finding the indicators for each learning objective template and matching the sentences or paragraph with this template taking into account that there is a degree of matching depend on the number of template indicators found, the number of nouns in the sentence, and some symbols and numbers, the final stage is to extract all the words from the text which fall in the same classification in the Arab Ontology tree, this classification can also be used to exclude certain words to improve the degree of matching. The methods which used to test and evaluate the accuracy of approach using calculate precision, recall and f-measure. The results show satisfactory results for some learning objectives like extracting definitions with precision = 95% and recall = 90%, while the results for other learning objectives were fairly results.

Keywords – Learning Objectives**,** Arabic Ontology, Bloom's Taxonomy

إستخلاص الأهداف التعليمية من النص العربي على أساس تصنيفات بلوم

إعداد: طارق عصام عبد العزيز علي

إشراف: د. بديع سرطاوي

الملخّص

تعد اللغة العربية واحدة من أكثر اللغات تعقيدًا وشمولية والحاجة إلى إيجاد نمط موحد ودقيق للتحليل وفهم اللغة أصبحت ضرورة ملحة في ضوء هذا التسارع التكنولوجي ، الذي يمكن أن يساعد في العديد من المجالات ، بما في ذلك التعليم والبحث ، وقد يكون ذلك مفيدًا في محاولة أتمتة التعليم باستخدام منهجية أهداف التعلم ونتائج التعلم.

الهدف من هذه الرسالة هو تحليل النص العربي من أجل استخلاص الأهداف التعليمية الهادفة من النص على أساس أول مستويين من تصنيف بلوم، التذكر والفهم باستخدام منهجين رئيسيين أولهما عن طريق بناء علم الوجود العربي (شجرة بدائية) في هذه المرحلة، والتي تتضمن تصنيف الكلمات العربية تصنيفًا ثابتًا وموحدًا يستند إلى أصل الكلمة، والمنهجية الثانية هي اعتماد نماذج قياسية لأهداف التعلم تشمل كل نموذج عدة كلمات رئيسية ومؤشرات ثم استخدام ثلاث مراحل. المرحلة الأولى تقوم بعمليات المعالجة المسبقة في النص مثل تقسيم النص إلى (فقرات، جمل، كلمات)، إزالة / تعيين علامات التشكيل ، استبعاد بعض الكلمات قبل البحث، المرحلة الثانية إيجاد المؤشرات لكل قالب من قوالب الاهداف التعليمية ومطابقة الجمل أو الفقرة مع هذا القالب مع الأخذ بعين الاعتبار أن هناك درجة من المطابقة تعتمد على عدد مؤشرات القوالب التي تم العثور عليها، وعدد الأسماء في الجملة، وبعض الرموز والأرقام، المرحلة النهائية هي استخراج كل الكلمات من النص الذي تقع في نفس التصنيف في شجرة الوجود العربي، يمكن استخدام هذا التصنيف أيضًا لاستبعاد كلمات معينة لتحسين درجة المطابقة.

الأساليب التي تستخدم لاختبار وتقييم الطريقة باستخدام حساب دقة النتائج حيث تظهر نتائج مرضية لبعض الأهداف التعليمية مثل استخراج التعريفات بدقة = 95٪ ونسبة إرجاع = 90٪ ، بينما كانت نتائج أهداف التعلم الأخرى نتائج معقولة..

الكلمات الأساسية – الأهداف التعليمية ، شجرة الأنطلوجيا العربية ، تصنيفات بلوم

Table of Contents

Declaration I

Acknowledgment II

Abstract III

الملخّص IV

Table of Contents V

List of Tables VIII

List of Figures IX

List of Abbreviations XI

Chapter 1 2

Introduction 2

1.1 Introduction 2

1.2 Statement of the Problem 3

1.3 Research question 4

1.4 Contribution 4

1.5 Scope and Limitations 5

1.6 Research Methodology 5

1.7 Research Outline 8

Chapter 2 10

Related Work 10

2.1 Literature Review 10

2.2 Theoretical Concern 12

Chapter 3 15

Theoretical and Technical Foundation 15

3.1 Ontology 15

3.1.1 Structure of Arabic Ontology 17

3.1.2 Contribution of ontology tree in the extraction of learning objectives 17

3.2 Global Terminology Recognition 18

3.3 Extracting Learning objectives based on Blooms Taxonomy 19

3.3.2 Knowledge Level Sections 22

3.4 Constructing Templates for Learning Objectives 23

3.5 Arabic Text Processing 24

3.5.2 Diacritical Marks 25

3.5.3 Original Word Extract..25

3.6 Performance Evaluation...26

3.7 Summary 27

Learning objectives Extraction Model..28

Chapter 4 29

Learning objectivesExtraction Model...29

4.1 The Basic Model 29

4.2 Constructing Ontology Tree 32

4.3 Collecting Data and Pre-processing. 33

4.3.1 Collecting and classifying Data 33

4.3.2 Pre-processing..35

4.4 Global Terminologies 37

4.5 Templates for Learning Objectives..38

4.5.1 Remembering level 38

4.5.2 Understanding level 40

4.6 Summary 42

Chapter 5 44

Implementation 44

5.1 Programs and Tools 44

5.2 Implement Pre-processing Stage..45

5.2.1 Splitting..45

5.2.2 Normalization 48

5.2.3 Tokenization 50

5.3 Ontology Based Extraction 50

5.4 Extracting Global Terminologies...52

5.5 Implement the General Templates to Extract LO 53

5.5.1 Remembering level 53

5.5.2 Understanding level 57

5.6 Summary...60

Chapter 6...62

Experimental Results and Evaluation 62

6.1 Experiments Setup 62

6.2 Preparation 63

6.3 Arabic Text Files and Scenarios 65

6.3.1 Sience Content. 65

6.3.2 Text content of history. 66

6.3.3 Text content of general topics. 67

6.4 Evaluation of Results 68

6.5 Discussion...73

6.6 Summary...74

Chapter 7...76

Conclusions..76

7.1 Summary...76

7.2 Future Work 77

References...78

Appendix: code to extract learning objectives 82

List of Tables

Table 3.1: Global Terminology Classification..18

Table 3.2: Learning objectives for each level.. 21

Table (4.1): Pronoun Classifications...34

Table (4.2): Huruf Classifications...34

Table (4.3): Splitting Indicators. 36

Table (6.1): Computer Specifications 63

Table (6.2): Summary of Precision, Rrecall and F-measure for All Text Contents 73

List of Figures

Figure 1.1: The Approach to Extract Learning Objectives..8

Figure 3.1: A Section of Ontology Tree – in English..16

Figure 3.2: A Section of Ontology Tree – in Arabic 21

Figure 3.3: Andersons Taxonomy...21

Figure 3.4: Model Flowchart for Arabic Text Processing 24

Figure (4.1): Model Flowchart for Extracting LO...32

Figure (4.2): Constructing Ontology Tree. 33

Figure (4.3): Pre-processing Stage...37

Figure (4.4): An example of Tokenization Process 37

Figure (4.5): General Template for Definitions..38

Figure (4.6): General Template for Enumerations...39

Figure (4.7): General Template for Comparisons..41

Figure (5.1): Original Arabic text before splitting...46

Figure (5.2): Code to Split Text into Paragraphs...46

Figure (5.3): Split the Text into Paragraphs...47

Figure (5.4): Code to Split Text into Sentences..47

Figure (5.5): Split the Text into Sentences 48

Figure (5.6): Code to Split Text into Words..48

Figure (5.7): Split the Text into Words..48

Figure (5.8): Removing Diacritics Marks..49

Figure (5.9): Placing Diacritics Marks...49

Figure (5.10): Extract the Original Word 50

Figure (5.11): Ontology Based Extraction...52

Figure (5.12): Code to Collect Global Terminology 53

Figure (5.13): Algorithm to Extract Definitions..54

Figure (5.14): Extracting the Enumerations..55

Figure (5.15): Extracting Verses of the Holy Quran..56

Figure (5.16): Extracting Comparisons...58

Figure (5.17): Extracting Cause-Effect...59

Figure (5.18): Implementation Flowchart...59

Figure (6.1): Results of Science Content 66

Figure (6.2): Results of History Content..67

Figure (6.3): Results of General Content. 68

Figure (6.4): Curve of Science Text Results..70

Figure (6.5): Curve of History Text Results 71

Figure (6.6): Curve of General Text Results 72

List of Abbreviations

LO	Learning Objectives
KU	Knowledge Unit
KL	Knowledge Level
KLS	Knowledge Level Section
LOT	Learning objectives Template
NER	Named Entity Recognition
DTM	Degree of Template Matching
OT	Ontology Tree
IDE	Integrated Development Environment
ML	Machine Learning
MVC	Model-View-Controller
GT	Global Terminologies
LU	Learning Unit

Chapter 1

Introduction

Chapter 1

Introduction

1.1 Introduction

Arabic is one of the most complex and comprehensive languages in the world. Its importance comes from being the official language of many countries in the world, in addition for being the language of the Holy Quran. Where the meaning of the word changes according to where it locates in the sentence and it also changes according to the diacritic marks on the word. This flexibility in the Arabic language imposed some kind of complexity in understanding the intended meaning for those who Arabic isn't their first language. This is what made learning this specific language a big challenge because it requires a deep understanding of the nature of Arabic text content. He also should know where are the most important sentences, concepts, terms and expressions in the text, and he should learn it optimally and effectively. The learner should also know what to learn from a certain learning activity, but this is a difficult operation, especially when we have a very large knowledge unit (book), which requires the learner to read the entire unit to determine the learning objectives. From here came the concept of computing the extraction of learning objectives [18].

Constructing an ontology tree for Arabic words will help a lot in text analysis. The classification of words into a group of entities based on the origin of the word and its common characteristics will play an important role in extracting learning objectives. By listing all the names mentioned in the knowledge unit that falls in the same classification within the ontology tree [2]. And also can help by excluding some classifications when extracting some learning objectives, which could enhance the accuracy of the results. In fact, constructing a detailed ontology tree which includes almost all Arabic words and all knowledge domains, it needs a great effort, and a lot of time. This research will focus on constructing the basic levels of the ontology tree, hoping that it will be fully constructed as future work.

The main useful information in the Arabic text, which may constitute the learning objectives was studied, and have been compiled into groups and adopted a general form for each group called the template. Thus the learning objectives template is a general template which was constructed based on the study of many diverse examples of various knowledge units and learning activity. And each of these templates has keywords which are called template indicators. For example, "the definition of terms" comes as a learning objective group that falls within the first level of Bloom's Taxonomy.The general template for definitions in Arabic. It consists of three main parts: the main term, indicators and the definition as shown in the following table:

Definitions Template	
The sentence	علم الوراثة هو العلم الذي يدرس الجينات
Parts of the template:	
1- The main term	علم الوراثة
2- The indicators	هو
3- The definitions	العلم الذي يدرس الجينات

The second group of learning objectives is the enumerations; also its general template consists of several parts as shown in the example in the following table:

Enumerations template	
The sentence	تتم عملية الري باحد الطرق التالية: 1- الري بالتنقيط 2- بالرش. 3- بالفقاعات
Parts of the template:	
1- The main term	طرق عملية الري
2- The indicators	التالية + 3/2/1
3- The Enumerations	الري بالتنقيط + بالرش + بالفقاعات

Next, the identification of the research problem and the objectives to be achieved, the importance of the thesis, scope and limitations and then the research methodology that followed to solve the research problem.

1.2 Statement of the Problem

Curriculum and scientific books in general that contain a lot of knowledge from the first level and the second levels of Bloom, cannot be inferred and derive those sections of knowledge without reading the book carefully. And the degree of focus and understanding vary from person to person. And when the book is read by more than one teacher each teacher will understand the book in his/her own way. This will make each teacher identify the learning objectives of the book based on his/her own understanding. In the end, we will find that the learning objectives identified by teachers are different, even if this difference is simple. And here came the idea of automating and unifying the process of extracting the learning objectives of the Arabic text, which I will try to prove in this research [20].

The problem of this research is the process of automating the extraction of effective learning objectives from the Arabic text.

1.3 Research question

To what extent this research will help in extracting effective Learning Objectives from Arabic text?

1.4 Contribution

The contribution that this research is expected to achieve is to automate and unify the process of extracting learning objectives from educational courses, in a systematic and consistent way to facilitate and speed up the process of extracting learning objectives. This is considered as an alternative to reading the entire book in order to extract the learning objectives, and also as an alternative to extracting different learning objectives for the same course, according to the different teachers.

In addition, starting to construct ontology tree will contribute significantly in the process of classifying the words of the Arabic language, in a uniform and systematic manner, according to the meaning and origin of the word. This will enhance the accuracy of the results in this research, and may help in future research [2].

In more details, the research will focus on the following contributions:

- Classifying knowledge in the first and second level of Bloom's taxonomy into groups, and then building a general form for each group called the template, which will be recognized through this template.
- Constructing an ontology tree for the Arabic words, based on the origin and nature of the words, which may open new horizons for future works.
- Building several algorithms for processing and analyzing Arabic text.

Thus, the contribution that this research is expected to achieve in more details is first by defining the learning objectives in the Arabic text and classifying them into groups by developing the keywords in the first and second levels of the Bloom's taxonomy and linking them to the learning objectives groups. For example, the keyword " define " in the remembering level, it is linked to the definitions as a set of learning objectives.

Second, the contribution of the Arab Ontology tree in the process of extracting learning objectives through the recognition of words, and based on them in the classification of words, it is considered a reliable source of classification of words, and thirdly, the development of some important algorithms in the processing of the Arabic language, such algorithms contribute effectively in the analysis of Arabic text.

1.5 Scope and Limitations

- The official curriculum will be adopted to extract learning objectives in the fields of science, Arabic text, religious culture and most of the curricula. Except for scientific curricula, such as mathematics and chemistry.
- The construction of the ontology tree will be at this stage simple and primitive.
- Non-Arabic words will be ignored in processing the Arabic text.
- Names of global international institutions and organization will be listed to be used as machine learning, in order to enhance the results.

1.6 Research Methodology

The methodology followed to achieve the objectives of this research as follows:

1- Studying the origins of the Arabic words, and constructing the basic levels of Arabic ontology.

A- Constructing the first level in the ontology tree, that contains the classification of the words of the Arabic language. The classification based on the science of existence and origin of the words, and based on the rule of the absence of synonyms in Arabic, and the root of this tree was the word "a thing" [2].

2- data collectionand identification of concepts and pre-processing:

The data are the Arabic language, and the collecting process will include the following steps:

A- Collect all global terminology, including global organizations, institutions, associations and groups, to be recognized when found in the text.

B- Collect Arabic words that may be needed from various sources and random texts, then divide these words according to the spaces between them to tokens and then classified.

C- To classify the word, type if it’s a noun or verb. A code is used to detect the past verbs according to the patterns of past verbs, which well known with its diacritical markings, and about the present and the future verbs will be derived from the past verbs by another code.

D- For other types of tokens such as pronouns, prepositions, vowels, numbers and symbols, I have manually listed and categorized them.

E- When the verbs pronouns hurrof, numbers, and symbols will exclude from the collected text, only the nouns will remain, and a large number of them will be categorized and linked to Arabic ontology tree.

F- Also, from the process of data collectioren, the Holy Quran was obtained as a text from the Internet, so all the verses were divided and classified according to the verse and verse number.

G- Divide each level of bloom (memory and understanding) into sections called knowledge sections or groups, for example, knowledge levels of remembering level are definitions. Enumerations, comparison and differentiation, cause and effect.

H- Adopt general forms called templates for each section of knowledge, to matching the text with these templates.

I- Adopting keywords for each section of knowledge, that will serve as indicators to indicate if found in the text, for example, the indicators of the knowledge section, such as definitions are in Arabic: "يعني ، المقصود ب ، هو مصطلح ", and for the knowledge section enumeration of things, the indicators are in Arabic "ثلاثة و جمع" or " اثنين/اثنان و مثنى".

J- Each learning objective is will be divided into two main parts: the question and the answer, or object and subject. For example, in the definitions section like "Global warming is a term used for the observed century-scale, rise in the average temperature of the Earth's", in which "Global warming" is the question and “is a term used for the observed century-scale rise in the average temperature of the Earth's" is the answer.

K- Pre-processing stage, which includes text splitting into paragraph sentences and words, Tokenization and text normalization like removing diacritical marks or setting diacritical marks.

3- : Extracting Learning objectives [4].

A- The text will be into:

- Paragraphs based on sentence dividers such as {'\t\n\n'}.
- Sentences based on sentence dividers such as (.).
- Words or tokens based on word dividers such as ('.', ' ', ';', ',').

B- The text is matched with the predefined knowledge groups templates and the concept of the matching degree is applied, which is a counter depends on the number of indicators and nouns in the text, and matching that has a high degree are selected.

C- As we mentioned earlier in some sections of knowledge, such as definitions, some terms are excluded from definitions such as persons’ names and country names, even if they came with indicators of definitions, here comes the role of the ontology tree.

4- : Extracting the single-edged words.

A- In this stage, all words within the same classification in the ontology tree will be extracted if they are more than one word.

5- : Extracting the global terms.

A- This process involves recognizing the global and standard terms in the text that has been defined and collected before.

6- : The evaluation:

A- At this stage, the accuracy of the results will be measured, and the learning objectives that have been extracted will be evaluated linguistically and logically [14].

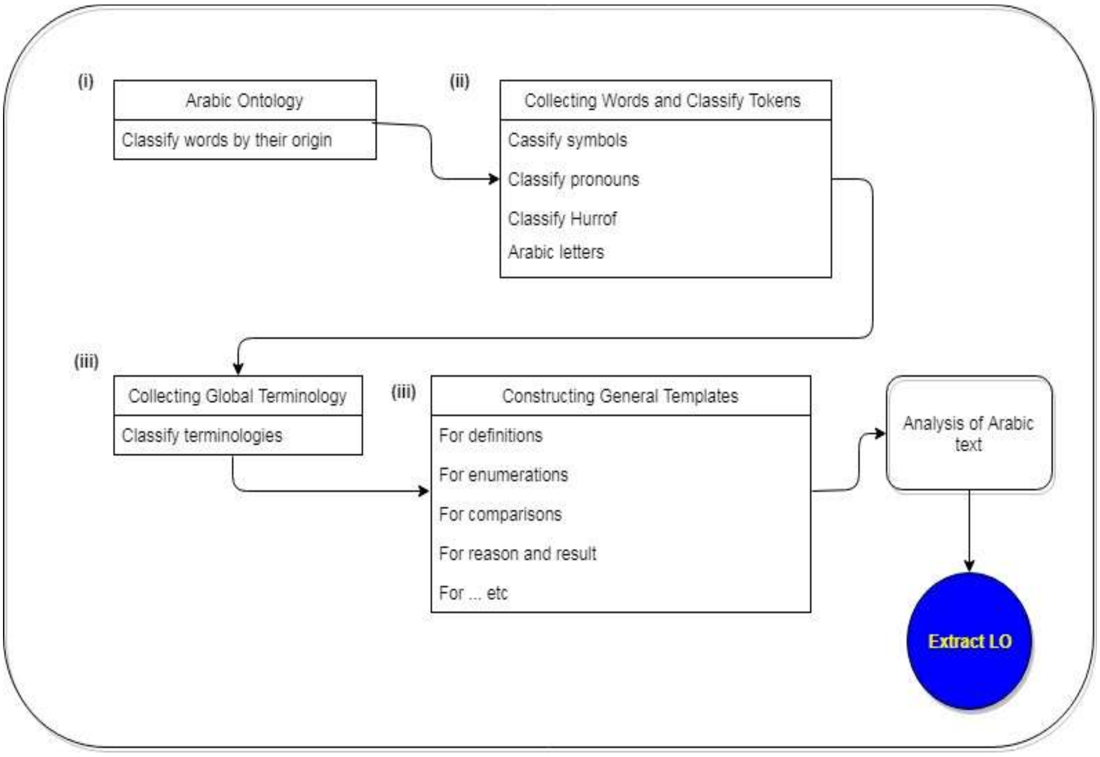


Figure (1.1): The Approach to Extract Learning Objectives

1.7 Research Outline

The research includes seven chapters; the first chapter an introduction and motivation addition to the specified basic research questions and contribution. Chapter 2 the background and related work and gives general definitions on the terms that will be used during the research. Chapter 3 is the theoretical and technical foundation. Chapter 4 describes the thesis methodology and approach that has been used; the methods of data collection and clarify the need to collect and classify them. It will also include a detailed description of the components of the methodology and the reasons behind adopting it methodology. Chapter 5 focuses on the implementation in details, Chapter 6 presents the experiments and the results, it will present the experimental design and all the related plans and work implemented within the experiments, afterward the chapter will show the results and the analysis on them using graphs and SPSS. Finally, the chapter describes other factors and experiments that affect the accuracy of learning objectives extracted from the various curriculums and present the results. Finally, inChapter 7 we will explain the conclusion and future work.

In the next chapter, we will study as much of the previous relevant work as possible in learning objectives, constructing the ontology tree, Bloom's taxonomy and text analysis.

Chapter 2

Related Work

Chapter 2

Related Work

2.1 Literature Review

For the intelligent control of an efficient and reliable Arabic language analyzing, this research has reviewed as many as possible of studies, one of the main researches Building a Formal Arabic Ontology [2]. The main idea for this study is to find a formal way of organizing the Arabic words. It includes putting words into categories and relating these categories to each other. The limitation of this research is the difficulty of including all Arabic language concepts from the sources of language, and linking the tree with word net [22]. The result of this research was semantic relations grows philosophically and logically disciplined without ambiguity.

Another research was an approach to assist learners to identify their learning objectives in the Personal Learning Environment (PLE) [3]. This approach enabled the learners to build their personal learning environment, defining their personal training courses and choose their learning activities. Through the use of the ontology, Bloom's taxonomy, and learner profile. But the limitation of this exploratory research was curriculum depends on several elements of existing programs. Which is likely to be provided by several institutions that exploiting different learning environments and learning tools in order to build three components: curriculum builder, learning activities recommended, and program profile manager. The result was building effective learning experience based on a clear learning objective, and a detailed curriculum describing the specific learning objectives.

Another approach that has gained popularity in the recent years is explored learning objectives on a large scale, through the concept of mapping of MOOC learner discussions [4]. This approach depends on learning objectives from Massive Online Open Courses MOOC, depending on comments mining provided by learners, then comments compared with LO, this means this exploratory research depend on MOOC from large scale knowledge and large growth questions, and in this regard, research suggests that MOOC practitioners and researchers should consider a multiple set of indicators success. And the result was conceptualized by Leximancer (data mining tool) represented in a chart, larger circles represent themes within the MOOC step comments, and dots represent smaller concepts.

And with regard to use Bloom's taxonomy as fundamental criteria for learning objectives, what is mentioned in the research "How to apply the Bloom taxonomy to software engineering" [1]. This paper has been reviewed in order to apply Bloom's taxonomy to software engineering, since it has tried to use Bloom's taxonomy to extract operational knowledge from software engineering, arguing that previous research focused on applying theoretical knowledge. This exploratory research led to two axes of expansions Knowledge area categorization and construct operational capability levels. Another important research "Student performance and Learning Outcomes in Higher Education Institutions" [5]. This research studied the previous strategy followed to measure the student performance to assess the students, according to acquired learning outcomes whereas the college has clearly stated learning outcomes and provide analysis of the student's results/achievements, and the differences, if any, and patterns of achievement across the programs /courses offered. And about Arabic text analyzing I've reviewed "Automatic Extraction of Ontological Relations from Arabic Text" [6]. This research proposes a methodology for automating the process of extracting ontological relations from Arabic text, by using Hearst's algorithm with additional enhancements for this purpose; such enhancements include pattern enrichment, pattern filtering, the application of negative patterns, and pattern evaluation. This is because the current Hearst's algorithm requires human intervention to extract a pure supervised activity. And also other related research "Analysis Model for Learning Outcomes (AMLO)" [7]. This study highlighted the importance of standards and criteria in improving the quality of higher education programs, and designed an analysis model for learning outcomes in a teacher education program.

In the Arabic text analysis area, I also reviewed "Intelligent Agent for Information Extraction from Arabic Text without Machine Translation" [8]. That employed sentiment analysis in order to create new books in the area of Islamic Hadith Narration automatically, instead of manual classification done before.

Another research is "A recognition system based on template matching for identifying handwritten Farsi/Arabic numerals"[28], this research only process handwritten Farsi/Arabic numerals, and it does not analyze entire Arabic sentences.

And from the research I have studied and which is related to the construction of the tree of ontology "Ontological tree Generation for Enhanced Information Retrieval"[31], this research extract information from big data and convert this information to nodes and edges in order to build hierarchical ontological tree.

And the research “Template Mining for Information Extraction from Digtal Documents”, explain the methodology of extracting information from the text based on template template that involves a natural language processing (NLP) technique.

The last search I've studied is "Arabic Character Recognition System Development"[29], this research uses algorithm called "Hilditch Thinning algorithm" to analyze the images and extract the latin and arabic characters from image using connected pixel component and building a template for this component consist od three parts main body, secondary object and noise, thes. Where the parts are determined depending on the size as the body has the largest size followed by the secondary body in size and what remains is noise. Such a research, though not analyzing Arabic text as text, but the idea of building a template to identify content is an idea that has been proven to be effective in analysis.

2.2 Theoretical Concern

From previous work, we note that some researches focused on the classification of Arabic words only, in such researches, there is no textual analysis to extract learning objectives. Other researches classified Arabic words and also analyzed the text to help the students to build their own learning objectives. Although these studies had worked to extract learning objectives, but such learning objectives had been specific to each student individually, and there is no formal and uniform way to extract static learning objectives. Other researches also focused on analyzing the user’s comments and free texts from online pages, then compared to LO. Such research relies on random texts instead of the formal curriculum.

We conclude from the previous research that none of them is specialized in analyzing official Arabic language texts, just analyzing some random comments and text, based on some keywords or classifying the words based on a previously defined word tree. It is not enough to adopt one of these methodologies alone to extract learning objectives from the formal educational curriculum, for example, classifying the words in the text is not considered a complete text analysis. And analyzing some comments and random texts from the web pages can be used to extract some information only, and can not be relied upon to extract learning objectives of the official curriculum.

Moreover, the process of extracting learning objectives from the formal curriculum requires high accuracy in text analysis, and static and uniform methods. And to achieve this, we must rely on several methodologies in analyzing the Arabic text, including classifying the words in a semantic way [2], and analyzing Arabic text based on template matching, this new methodology works on the importance of this study that it built for the main information in language and matching text with these templates [23].

Chapter 3

Theoretical and Technical Foundation

Chapter 3

Theoretical and Technical Foundation

In this chapter, we will explain the mechanism of constructing Arabic ontology tree and its components, and then the development environment that will be used in the experiment will also be discussed. Finally, the method used to evaluate the results and evaluate performance will be highlighted.

3.1 Ontology

Ontology tree is a tree that includes the classification of the original words in Arabic based on the origin of the word. This tree starts from the word "thing" as the root of this tree and then branch out depending on what this thing might be. Each node in the tree may be a class or a branch of its parent node or part of it [2].

The following is a set of definitions of the concept of Ontology:

- **T. R. Gruber**: “ontology is a formal explicit specification of a shared conceptualization” [9].
- **H. Herre et al**.: “Formal Ontology is the science that is concerned with the systematic development of axiomatic theories describing forms, modes, and views of being at different levels of abstraction and granularity.” [10], (The ontological argument for God’s existence, as stated by **Anselm of Canterbury** [1033–1109] and others, asserted that the Supreme Being was the highest term in a scale of terms ranging from the lowest form of being to infinity, referred to as the “Great Chain of Being.” So if we use modern ontology language to describe an Anselm’s argument, God is the ultimate Thing class, and all lower beings, from the angels down to the microbes, are nested subclasses of Thing.) [11].

Ontology means the “science of being”. Ontology is the most comprehensive of all sciences, insofar as it covers everything that exists. I will go into details later on, but for now: Among philosophers, there is no commonly agreed concept of the ontology. The interpretations of philosophers differ just like the multiple ways of looking at “being” [12].

G. Pickert: “An Ontology is a seven-tuple, O = (L, C, R, F, G, H, A)” [13].

With:

C: A set of concepts

R: Set of binary relations on C (Relationships between concepts)

F: Function connecting symbol sets to sets of terms

G: Function connecting symbol sets to sets of relations

T: Taxonomy for the partial ordering of C in generic concepts / narrower concepts

A: Set of axioms (constraints): statements with elements of C and R.

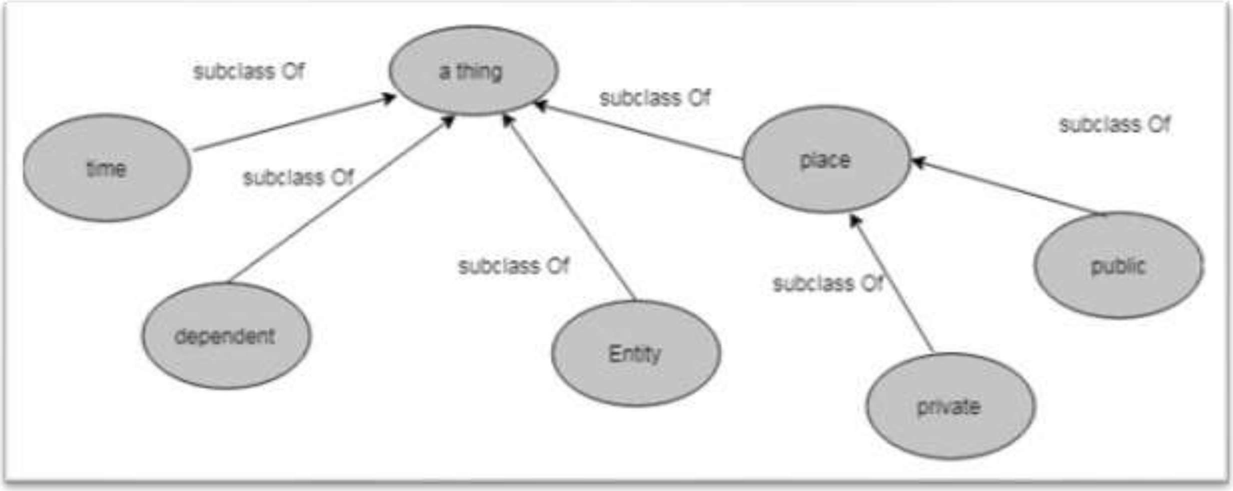


Figure (3.1): A Section of Ontology Tree – in English

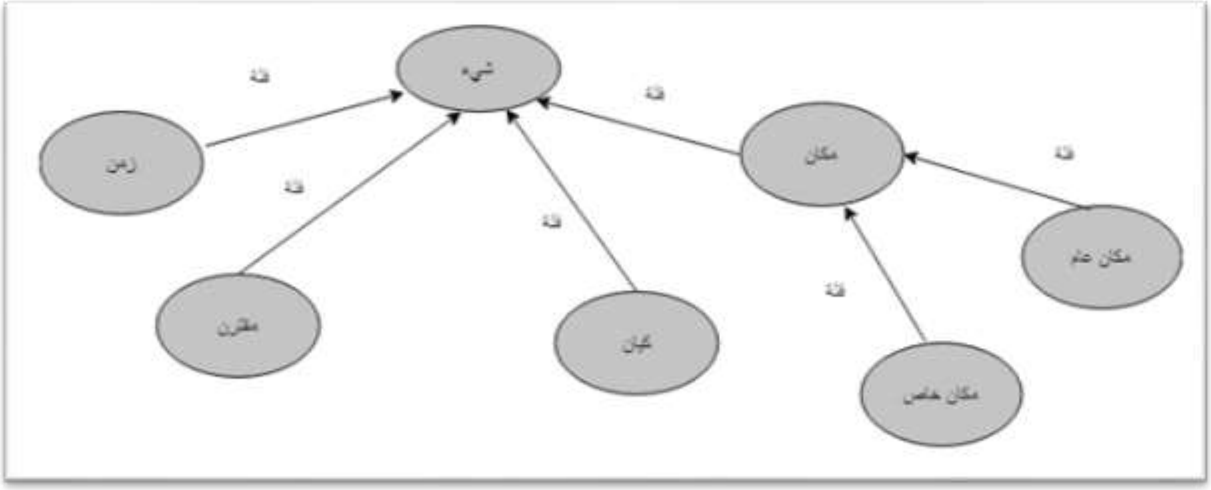


Figure (3.2): A Section of Ontology Tree – in Arabic

According to the definition of ontology:

Ontology tree is a tree that includes the classification of the original words in Arabic based on the origin of the word. This tree starts from the word "thing" as the root of this tree, and then branch out depending on what this thing might be [2].

We note from the above figure that the concept (**thing**) is the root of ontology tree, and the sub concepts ("**time**", "**place**", "**entity**", "**dependent**") are what the concept (thing) might be.

3.1.1 Structure of Arabic Ontology

The ontology tree consists of individual words of the Arabic language, including what is tangible, such as animals and plants, and what exists virtually and cannot be seen, like angels and energy, and including words that are linked to other assets [24], so that they have no importance without them such as attributes, events and abstract words, in addition to the words that indicate time. Each word in the tree is either a class of its parent word or part of it [2], For example, trees, grasses and vegetables are varieties of plants while flower, stem and stems are part of it, There is also confusion in the distinction between descriptions and characteristics in most of Arabic dictionaries , but this tree distinguishes them since the characteristics belong to the object and the body of the thing itself, while the descriptions are closest to the description of the thing by sight, For example, when we say hardness, we describe the body of the thing and it be common in the same type of thing, but when we say beauty, we describe the current scene.

3.1.2 Contribution of ontology tree in the extraction of learning objectives

Ontology tree will contribute in the extraction of learning objectives through two basic principles, the first one by grouping all the words in the text which have the same classification in the tree, and the second one by excluding the words of some classifications during analyzing the text to extract accurate learning objectives. For example, excluding countries and person’s names when extracting definitions.

In more details, while processing the Arabic text and splitting the text into words, the words are compared with the tree of the ontology and compiling all the words that have the same classification in the tree of the ontology in groups, especially if the number of words for the same classification in the group is more than two, such these words may be a list of a particular category, types or specific characteristics and even if they mentioned between the lines and did not come in the form of points listed.

Formal Definition for extracting the concept from the Ontology tree (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 ⊂ B. And if and only if all elements in A belongs to the set B and every element in B belongs to set A, i.e., A ⊆ B and A ⊇ B.

Definition 2: If A element in a set B Then A belongs to B, i.e., A B.

3.2 Global Terminology Recognition

Because the ontology tree includes only individual Arabic words, the standard terminologies (which also called Global terminologies or general terminologies) will be grouped together and classified in a data repository, according to various fields of science, such as medicine, politics, economics, history, physics, mathematics, technology, sports and biology, it's contribute significantly in the extraction of predefined terminologies from the Arabic text, and works on the basis of the search for the terminology in the field of a particular science determined by the field, in which the text speaks, such as the names of organizations and institutions as shown in Table (3.1), which in the end have a classification in the ontology tree under an individual word.

Table (3.1): Global Terminology Classification.

Global Terminology (Arabic)	Global Terminology (English)	Keyword	Science Field
منظمة حقوق الإنسان	Human Rights Organization	Organization	Politics
نقص المناعة المكتسبة	AIDS	disease	Medicine
إكتفاء ذاتي	Self-sufficiency	economy	Economy

3.3 Extracting Learning objectives based on Blooms Taxonomy

Definitions of some key terms

Knowledge unit: is a term used for any written text in the official Arabic language, whether it is a full curriculum or part of a curriculum.

Knowledge level: is the level at which ideas, phrases and points are embedded, which have been learned from the course and this level as classified by Bloom can be: knowledge, comprehension, application, analysis, synthesis and evaluation.

Learning Objectives: are statements that define the expected goal of a curriculum, course and the lesson of demonstrable skills or knowledge that will be acquired by a student as a result of instruction, also known as: instructional objectives, learning outcomes, learning goals.

Knowledge Section: Is the classification of ideas and points that can be learned from the text at each level of knowledge into groups.

3.3.1 Bloom Taxonomy

It is a set of classifications of learning objectives developed by (Benjamin Bloom), in 1956, which are as follows: Knowledge, Comprehension, Application, Analysis, Synthesis and Evaluation.

And in 2001 Lorin Anderson and David Krathwohl Revised Bloom's Taxonomy, and the knowledge levels became as follows:

1- **Remember**:
   - Recognizing.
   - Recalling.

2- **Understand**:
   - Interpreting.
   - Exemplifying.
   - Classifying.
   - Summarizing.
   - Inferring.
   - Comparing.
   - Explaining.

3- **Apply**:

  - Executing.
  - Implementing.

4- **Analyze**:

  - Differentiating.
  - Organizing.
  - Attributing.

5- **Evaluate**:

  - Checking.
  - Critiquing.

6- **Create**:

  - Generating.
  - Planning.
  - Producing.

They replaced Bloom’s nouns with verbs to reflect the nature of thinking for each category, also they switched the last two stages of Bloom’s Taxonomy on the grounds, that the learner’s ability to evaluate comes before his ability to create/synthesize, And the last Revision was by adding a fourth knowledge dimension to the first level, whereas, in Bloom's Taxonomy, the Knowledge level includes three knowledge dimensions: factual (basic elements of knowledge), conceptual (the interrelationships between basic elements of knowledge), procedural (the “how-to” part of knowledge) and with Anderson the dimensions: factual, conceptual, procedural: metacognitive (knowledge of cognition and awareness of one’s own cognition).

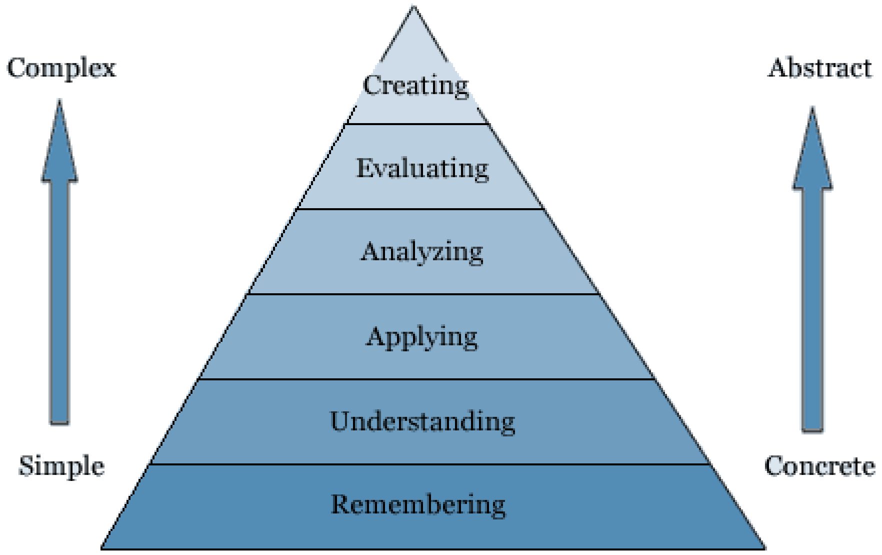


Figure (3.3): Andersons Taxonomy

Remember: recall facts and basic concepts.

We might ask students to recite something you've taught them, quoting information based on previous lectures, reading material and notes. learners can use verbs like define, describe, identify, label, list, outline, recall, and reproduce to effectively measure success in this level also using simple questions and answer periods, or multiple choice questions.

Understand: classify, describe, discuss, explain, indentify, locate, recognize, report, select, translate.

In this level students able to discuss a problem or idea in their own words. To measure this level by verbs like defend, explain, generalize, paraphrase, summarize and translate.

Apply: execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch.

The student will now have to take what they've learned and apply it to a scenario outside of the classroom. Verbs to use in this stage of Bloom's taxonomy include apply, demonstrate, predict, show, solve or use.

Analyze: differentiate, organize, relate, compare, contrast, distinguish, examine, experiment, question, test.

students can draw connections between ideas, utilize critical thinking, and break down knowledge into the sum of its parts. This can include using logical deduction to figure out how a piece of equipment works. Key verbs for measurement include analyze, break down, compare, contrast, differentiate, deconstruct and infer.

Evaluate: appraise, argue, defend, judge, select, support, value, critique, weigh.

The students make an educated judgment about the value of the material they've just learned, applied and analyzed, to be able to tell the difference between fact and opinions or inferences. That could include finding an effective solution to a problem. The verbs used in this level like Appraise, conclude, critique, evaluate, support and summarize.

Create: design, assemble, construct, conjecture, develop, formulate, author, investigate.

The students demonstrate their knowledge by applying what they've learned, analyzed and evaluated, and building something, either tangible or conceptual. like writing a manual or report on a particular topic. Verbs to use include categorize, combine, compile, devise, design, generate, modify and write.

Starting from the bottom level, and then ascending to higher levels, because the bottom levels require less cognitive processing but provide and important base for learning, while the higher levels require greater degree of cognitive processing with deeper learning and can only be achieved once the lower-order skills have been mastered, also their evaluation and implementation is somewhat difficult at this stage and therefore this research will focus only on the first and the second level of Bloom's taxonomy.

3.3.2 Knowledge Level Sections

According to the first two levels of revised Bloom's Taxonomy (Remember, Understand), each of these levels is divided into subcategories called sections, the sections for remember level as follows: Recalling which includes Definitions and enumerations. And recognizing includes retrieving facts, vocabulary, concepts and general terms**.** And the sections of understanding level: Interpreting, Exemplifying, Classifying, Summarizing, Inferring, Comparing and Explaining as shown in Table (3.2).

Table (3.2): Learning objectives for each level.

Bloom Level	Level Section	Learning Objectives
Remember level	Recalling	Definitions, Enumerations
	Recognizing	Retrieving facts, vocabulary, concepts, general terms
Understand level	Comparing	Comparison,
	Inferring	Cause and Result
	Summarizing	
	Classifying	

3.4 Constructing Templates for Learning Objectives

Because The templates are gaining more and more importance due to the fast growth of digital information resources. The templates are considered the general form for the main information in the Arabic text objectives, that contains indicators and keywords that indicate it.

Many studies in the analysis of texts and images have been based on the methodology of templates and have proved effective in predicting [23] [29]. For example, using templates in the compilation of research and articles from electronic newspapers and social networking sites by building a template consisting of the author and title and body. Also templates are used in web search engines-such as Alta Vista-and in meta-search engines-such as Ask Jeeves-for helping end-users generate natural language search expressions [32].

And several systems were developed to retrieve information in specific areas, such as news/fact retrieval and patent using the template mining approach that involves a natural language processing (NLP) technique to extract data directly from text form recognizable patterns, when text matches a template, the system extracts data according to instructions associated with that template from the text with standard format [32].

Therefore, we built these templates by studying various Arabic texts from various fields, and studying the general form of each group of information, and compiling these groups, then constructing their templates according to their general form. Each template consists of the following three main parts: The object or the main term, indicators referring to this term and the subject or the answer.

For example, the template for Definitions is:

- The object: noun word (Single / Compound).
- The indicators in Arabic: (يعني / هو / المقصودب/ هومصطلح).
- The subject: Is a sentence supplement.
- The question verb: (يعرِّف).

The general template for enumerations is:

- The object: plural noun word (Single / Compound).
- The indicators in Arabic: (التالية/ الآتية/ عددكتابةثلاثةأوأربعة).
- The subject: A set of subsequent sentences that begin with a symbol or an increasing number or letters arranged alphabetically.
- The question verb: (يعدِّد).

And the general template for comparisons is:

- The object: noun (Single / Compound).
- The indicators in Arabic: (الفرق بين / الإختلاف بين).
- The subject: Is a sentence supplement.
- The question verb: (يقارن/يفرِّق).

And the general template for inferring is:

- The object: noun (Single / Compound).
- The indicators in Arabic (أدالى/بسببذلك /لذلك / لذلكالسبب/ نتيجةل / نتجعنه/ سبّب / لهذاالسبب).
- The subject: Is a sentence supplement.
- The question verb: (يعلِّل).

3.5 Arabic Text Processing

In this section, we will discuss the methodologies and the tools used to deal with Arabic text, and the diagram of the Arabic text processing as shown in Figure (3.4).

3.5.1 Tokenizer

A special script to split text based on whitespace in the text, this script divides the text into paragraphs and sentences and divides each sentence into units based on the whitespace these units called tokens and each token may be:

- Word: Arabic word whether with diacritical marks or without it like (noun, verb - past/present/future, pronoun, harrf).
- Number.
- Symbol: and we classified symbol into several types, for example math symbol ('+', '-', '/'), keyboard symbols ('@'), text symbols ('!', ',', '.') and currency symbols (‘$’, ‘£’) and symbol (e.g. ‘&’, ‘^’).
- Letters: The letters that make up the Arabic language in order, whether in abjadic order or alphabetical order.
- Others: Anything not mentioned above like foreign characters.

3.5.2 Diacritical Marks

We also wrote another script to remove the diacritical marks for the Arabic words when necessary, but to apply the opposite and put the diacritical marks on the words if that’s necessary, we wrote another script to connect to (Multiple API), Multillect it is a web-based application to assign diacritical marks on Arabic words.

3.5.3 Original Word Extract

In some cases, we need to find the original word by removing prefixes and suffixes character/s from the word to extract the original word (root), so a special script has been written to perform this task.

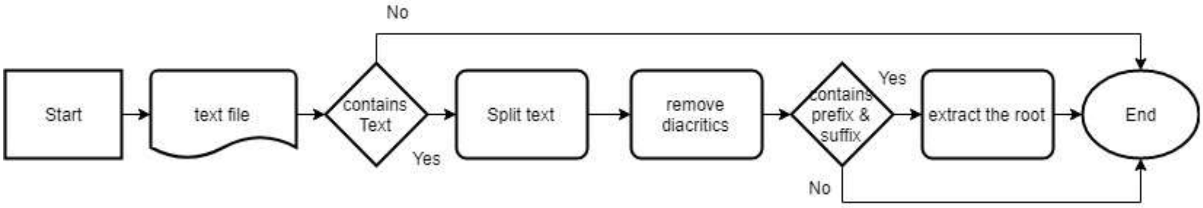


Figure (3.4): Model Flowchart for Arabic Text Processing

3.6 Performance Evaluation

Performance evaluation includes measuring the extracted learning objectives of the knowledge unit (LU), by using several performance metrics like accuracy, precision, recall, and F-measure. Such metrics used to evaluate the results when the experiment is concerned with qualitative results more than quantitative results [19].

Recall or the Sensitivity (as it is called in Psychology) is the proportion of Real Positive cases that are correctly Predicted Positive. This measures the Coverage of the Real Positive cases by the +P (Predicted Positive) rule. Its desirable feature is that it reflects how many of the relevant cases the +P rule picks up [25].

Precision or Confidence (as it is called in Data Mining) denotes the proportion of Predicted Positive cases that are correctly Real Positives [25].

The difference between accuracy and precision is: accurately describes the difference between the measurement and the part's actual value; while precision describes the variation you see when you measure the same part repeatedly with the same device [15].

If we denote the correct extracted learning objectives to (TP), and correct not extracted is true negative extracted (TN), false extracted (FP), false not extracted (FN) and realistic learning objectives that already exist in the LU (TP + TN + FP + FN).

- **TP: True Positives**: These are the learning objectives correctly predicted, and they are the actual learning objectives.
- **TN: True Negatives**: Correctly predicted as not learning objectives and they are not the actual learning objectives.

False positives and false negatives, when the actual learning objectives contradict with the predicted learning objectives

- **FP: False Positives**: means, predicted as correct learning objectives, but they are not the actual learning objectives.
- **FN: False Negatives** (missing points): means, predicted as not learning objectives, but they are the actual learning objectives.

The accuracy also means the percentage of results obtained as a result of our work and the actual results already exist [27].

So the equation formula for **accuracy** = **(TP + TN) / (TP + TN + FP + FN)**.

Precision also means: measure the same factor multiple times and then note the difference in the results in each measurement process, Greater precision means less variation between measurements [27].

Precision = **TP / (TP + FP)**.

= | relevant Learning objectives **intersect** extracted Learning objectives| / | extracted Learning objectives|**.**

Recall: The number of correct learning objectives that have been extracted is measured against the actual learning objectives in LU [27].

Recall = **TP / (TP + FN).**

= | relevant Learning objectives**intersect** extracted Learning objectives| / | relevant Learning objectives|.

F-measure or balanced F-score (F1 score) is the harmonic mean of precision and recall:

F-measure = **2* precision * recall / precision + recall**.

And for precision = **(TP + TN) / (TP + TN + FP + FN)**.

3.7 Summary

We explained the basic model on which this research is based, such as constructing Arabic ontology tree, and the classification of global terms, according to the various fields of science, in order to recognize and extract these terms once they are mentioned in the Arabic text, then we presented the revised Bloom's taxonomy and detailing each level, after that we presented the method for building templates for each of these levels, and explaining the mechanisms that have been used to process and deal with the natural Arabic text. Finally, we explained the methods that will be followed to evaluate the extracted results. In the next chapter, we will study and explain each component of the basic model separately.

Chapter 4

Learning objectives Extraction Model

Chapter 4

Learning objectivesExtraction Model

In this chapter, we present Learning objectives extraction model and our approach in order to extract meaningful learning objectives from Arabic text based on the basic model which include Arabic ontology tree, learning objectives templates and extracting the global terminology. In the beginning we will give an overview summarizing the approach--. Then we will explain extracting the words from Arab Ontology, compiling data and concepts identification, extracting learning objectives based on general templates, extracting the global terminology from the Arabic text. Finally, we will show the importance of each stage in contributing to the overall process in order to reach the results.

4.1 The Basic Model

The learning objectives that can be extracted from the Arabic text within the first and the second level of Bloom's taxonomy (remembering and understanding) can be random names and terms that are mentioned between lines without a fixed and uniform pattern.

For example, a group of animals or plants or metal elements are mentioned between the lines separated by a comma, such classificationscan not recognized by analyzing the general form of the sentence in which they were written, but in order to be recognized, we must have a prior classification of such words that contains a clear sequence based on the origin of the words, hence came the idea of constructing Arabic ontology, which made up of single Arabic words that are classified according to the origins of the words in the existence, reality, as well as the basic categories of being and their relations, for example, A word creature can branch out in the tree and can be a human being, an animal, a plant, etc., .

But since the ontology tree contains only single words and some of these terms and classifications that come in the Arabic text can be compound terms such as scientific terms, names of international organizations, agricultural terms and others, and such these terms have to be kept separately from the ontology tree, and categorizing them according the fields and the meaning of these terms with a keyword so that this keyword must be also classified within the ontology tree, these are known in our model as global terms which are interrelated with the ontology tree through the keywords for each set of global terms.

The Arabic text can also contain information that comes in a uniform and almost constant pattern in linguistic form and style such as the definitions, listed points, the comparisions and the cause and effect. For example the Arabic text may contain a definition of some terms, such definitions, wherever mentioned in the text, if we look at them, we will notice that they have the same formula, and it is as follows:

“The terms is ...”, the terms , indicator and the definition, these three basic parts are repeated in most definitions, and the same way regarding the listed points where it consists of two main parts : the basic part which consists of the terms and plural and some indicators, and the secondary part which consist of listed sentences whether were sequential or nonsequential, this is the general formula for the listed point, as well as for comparisons and cause-effect. So each of them has a general formula and a particular pattern that distinguish it from others in the text which leads us to think about building general templates for each of these knowledge points to be recognized from the text by matching the text with these templates, but bearing in mind that each of these templates has its own rules, indicators and constraints.

Therefore, the idea of building general templates to extract the learning objectives is not only a process of comparison and matching in the data store, but will be a process of analysis depending on other sub-processes like text processing and words classification and comparison and extraction of semantic relationships and this requires a lot of accuracy and comprehensiveness in building templates, because it will have a major role in extracting learning objectives.

The Arabic text can also contain constant information that does not change at all. Such information may differ from the concept of well-known global terms, as it may not be terms but rather complete and fixed texts whose letters do not change wherever they are mentioned, Examples of such information are: verses of the holy quran, historical and religious information such as books of Hadith, and poems of poetry. Dealing with such information will be no more than comparing it with the database and presenting it in order to enhance the extracted learning objectives.

Therefore, this is the type of information that any Arabic text may include, which is focuses on the first and second levels of Bloom's taxonomy, remembering level and understanding level, and extracting each type of information by using a different methodology and constraints. As an example to extract the classification of the original words, a well-defined and well-studied ontology tree must be constructed to be relied upon in extraction process, and in order to recognizing global terminology in the arabic text, we must collect as much as possible of global terms to be categorized and linked to the ontology tree and the same way regarding other static information, but for dynamic information of a similar and unified style, as we mentioned earlier in order to recognize them in the text, the methodology of constructing general templates must be thoroughly and comprehensively adopted, so these methods and constraints will work together within integrated basic model, in order to achieve the same goal of trying to extract meaningful learning objectives, and this basic model works as shown in Figure (4.1).

But we should know that the implementation of any of the methodology mentioned just above must be interspersed with some partial processes that contribute to the analysis of the text, which include splitting the text, removing diacritics marks from the text, checking the nouns words and checking the plural words, such processes, even if they are simple processes but the basic model relies on them mainly to achieve the main goal.

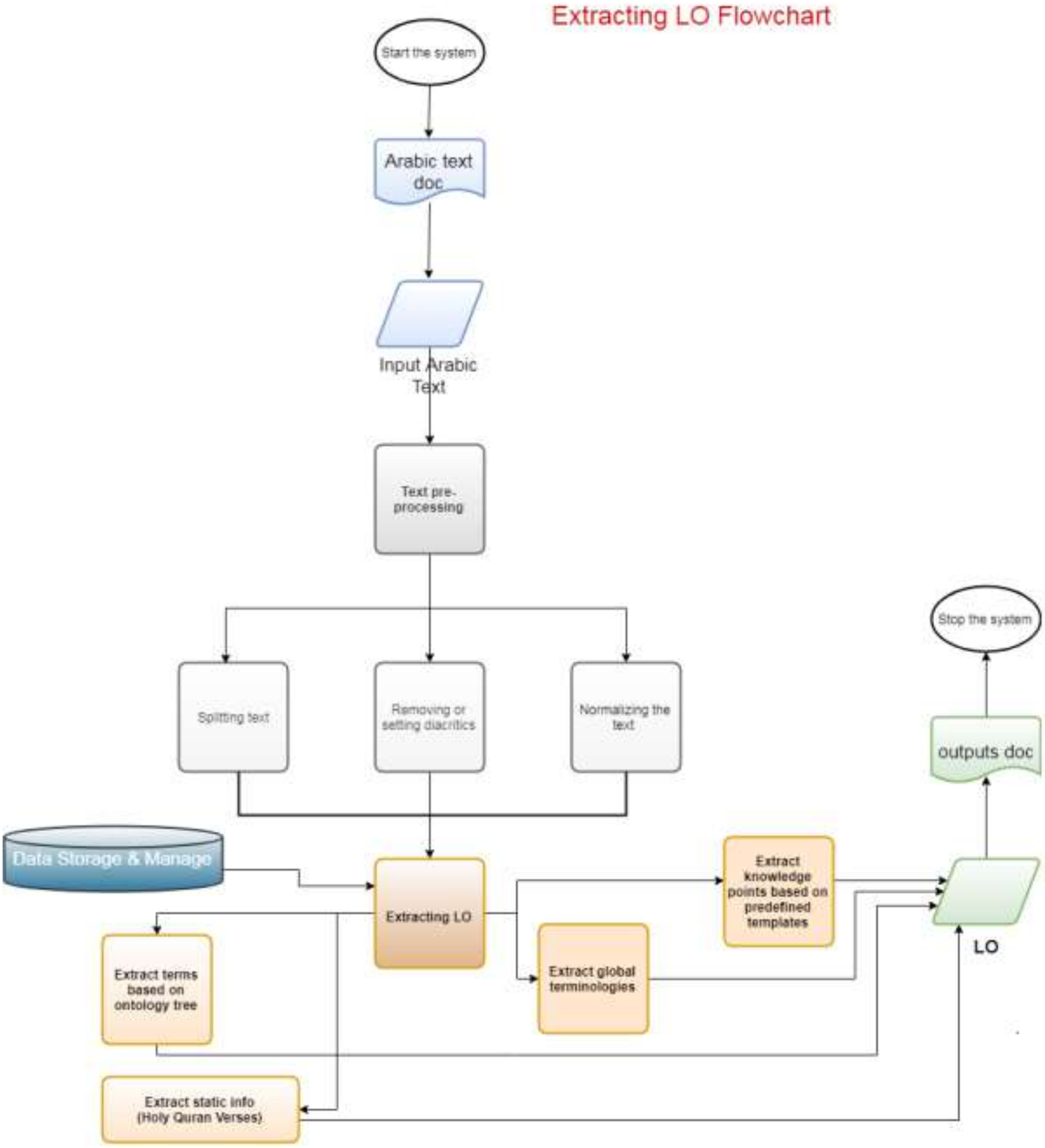
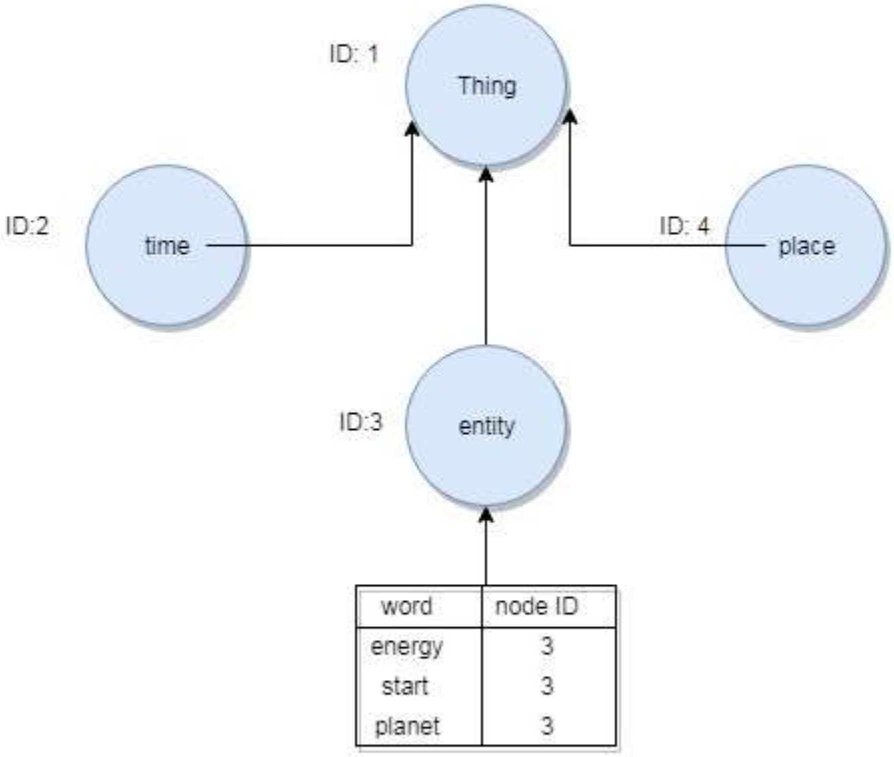

Extracting LO Flowchart
Start the system
Arabic text doc
Input Arabic Text
Text pre-processing
Splitting text
Removing or setting diacritics
Normalizing the text
Data Storage & Manage
Extracting LO
Extract knowledge points based on predefined templates
Extract global terminologies
Extract terms based on ontology tree
Extract static info (Holy Quran Verses)
LO
outputs doc
Stop the system

Figure (4.1): Model Flowchart for Extracting LO

4.2 Constructing Ontology Tree

The method of constructing the ontology tree is to start from the root word and branch out by the word classifications and what this single word can be and we assign an ID for each node in the tree after that when the words are collected either they are nouns or verbs all the words falling under the same node word in the tree take the same node ID.



word	node ID
energy	3
start	3
planet	3

Figure (4.2): Constructing Ontology Tree.

The figure above means in the ontology tree we define an ID for each node, and any number of words falling under this node when the words are compiled takes the same ID number, for example, a “plant” word has an id number = 101, and when all plants are compiled into the Arabic words table, they are given the same ID number as a foreign key.

4.3 Collecting Data and Pre-processing.

The data collection process involves collecting and classifying all that the text may contain such as Arabic words, verb, pronouns, numbers, letters and symbols...etc, and Pre-processing include splitting the text into paragraphs, sentences, words also include text normalization like assign or removing diacritical marks from the text.

4.3.1 Collecting and classifying Data

This process includes collecting and classifying the following:

- The Construction Alphabet: collecting and sorting the Arabic building letters in alphabetical order and Abjad order.

- The symbols: collect all the symbols that may appear in the text where they are divided into three types: text symbols, mathematical symbols, keyboard symbols.
- Numbers: the numbers from 1 to 100,000 were arranged and converted by script to words.
- Pronouns: pronouns in Arabic were collected and classified into types and subtypes, types like separated pronoun, connected pronoun, demonstrative pronoun and relative pronoun (ضمير منفصل ,ضمير متصل ,ضمير الاشار ,الاسماء الموصولة), and sub types like speaker, addressee, absent, close, far, specialist, common (متكلم, مخاطب, غائب, قريب, بعيد, متخصص, مشترك) as shown in Table (4.1).

Table (4.1): Pronoun Classifications.

Pronoun Type	**Pronoun**
separated	أنا / نحن/ أنتما/ أنتم
connected	هما / هم/ هن / وا
demonstrative	ذلك / ذاك / تلك / هنا / هناك
relative	اللواتي / اللذين

- Huruf: Huruf in Arabic were also collected and classified into several types like (Huruf alnaffi, Huruf ALnasb, Huruf Aljazm, Huruf Alatef… etc) in Arabic (حروف النفي, وحروف النصب حروف الجزم, حروف العطف .... وهكذا) as shown in Table (4.2).

Table (4.2): Huruf Classifications.

Huruf Type	**Huruf**
Alnaffi	ما / لا / لم
Nasb	ان / لن / كي
Aljazm	لما / لم / لا
Alatef	و / ف / ثم / حتى / أم / أو / بل / لكن

- Verbs and Nouns: Because of the scarcity of resources that contain all the Arabic words and was difficult to collect them because of the large number of words even in the famous Arabic dictionaries was not possible to obtain data from these dictionaries because they are picture pages so, we have collected the words from the lexicon of "لسان العرب" by splitting the text into tokens and excluding all that is not a verb or a noun

  like pronoun, Hurruf, symbols ... etc, and through a script written to distinguish the past verb from these words depending on the patterns of the past verb and also derive the present and future verbs from the past verb, and in this way about 8,900 verbs were collected, all the patterns of past verbs are as follows:

  ”فَعَلْ", "أَفْعَلْ", "فَعَّلْ", "فاعَلْ", "إنْفَعَلْ", "تَفَعَّلْ", "تَفَاْعَلْ", "إسْتَفْعَلْ", "فَعْلَلْ", "تَفَعْلَلْ", "إفْعَلَّ", "إفْعَاْلَّ"

And then all the remaining words are nouns and during the analysis of the Arabic text we also need to know the classification of words in terms of singular and plural. Therefore, we must classify the nouns on the basis of singular and plural and derive all plural nouns in order to achieve this I have built an algorithm that compares all patterns of plural as the following:

أَفْعُلٍ", "أَفعالٍ", "أَفْعِلَةٍ", "فِعْلَةٍ","فُعْل", "فُعُل", "فُعَل", "فِعَل", "فُعَلة", "فَعَلة", "فَعْلى", "فِعَلة", "فُعَّل", "فُعَّال", "
"فِعال", "فُعول", "فِعْلان", "فُعْلان", "فُعَلاءُ", "أَفْعِلاءُ","فعالِل", "فعاليل", "أَفاعِل", "أَفاعيل", "تفاعِل", "تفاعِيل",
"مفاعِل", "مفاعيل", "فواعِل", "فواعيل", "فعائل", "فَعالى", "فَعالِي", "فُعالى", "فعاليٌّ".

{جمع المذكر والمؤنث السالم}.

In this way we can distinguish the word if it is singular or plural, and if we want to distinguish the word is male or feminine through some indicators in the word if the last character is "ة" or "ى" or the word end with "اء" so it's feminine else male.

4.3.2 Pre-processing

After collecting data, the role of pre-processing come and it is considered one of the most important stages in our approach because they greatly affect on the results, this stage is intended to prepare the Arabic text to be input before extracting learning objectives, and this stage includes several sub-processes each one of them is considered a specific script to accomplish a specific function, these sub processes are: Splitting, Normalization and Tokenization which shown in Figure (4, 3).

- Splitting:

  The splitting includes splitting the text into paragraphs, sentences and words depending on tokens indicators in the text for each of them and there are indicators separating the paragraphs and indicators separating the sentences and indicators separating the words and this is necessary to know the boundary of paragraph and sentence to prevent exceeding the range of paragraph or statement when we analyze the text in order to extract learning objectives, these indicators shown in Table (4.3), but with regard to the indicators that divide the sentences sometimes we may exclude comma from the indicators if we need to get the whole sentence until the dot.

Table (4.3): Splitting Indicators.

Paragraphs	**>= New Line * 2**
Sentences	(";", ".", "؟","؛", "?", "!", "،")
Words	**(" ", "|", "\\", "/", ".", ";", ",", "؛", "?", "!", "(", ")", "،", "[", "]", "{", "}", "\"", "\'", "\t", "\n", "\r\n", "\n\n")**

- Normalization: is an attempt to unify the text and make it as close as possible to the general form of words and this process includes several procedures such as follows:
  1. Removing diacritic marks.
  2. Placing diacritic marks: it is API I programmed it and connected to the site "multillect.com" that take Arabic text and return the text with diacritic marks.
  3. Removing excess letters from the original word like "ال" or "ة" or Prepositions…etc.
  4. Removing numbers, symbols and special characters from the word.
- Tokenization: it is a fundamental process in the pre-processing stage to split the text based on the space only, where the output is a list of string called tokens, and each token may be a number, symbol, word, or special character as shown in Figure (4.4).

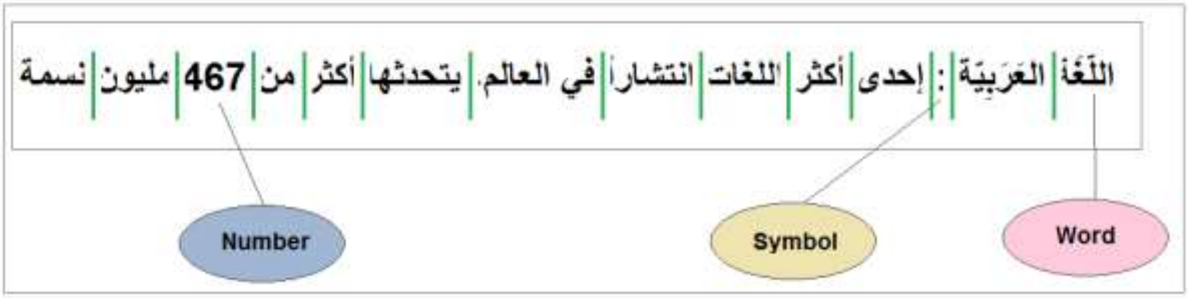

اللّغة العَرَبيّة : إحدى أكثر اللغات انتشاراً في العالم. يتحدثها أكثر من 467 مليون نسمة
Number
Symbol
Word

Figure (4.4): An example of Tokenization Process

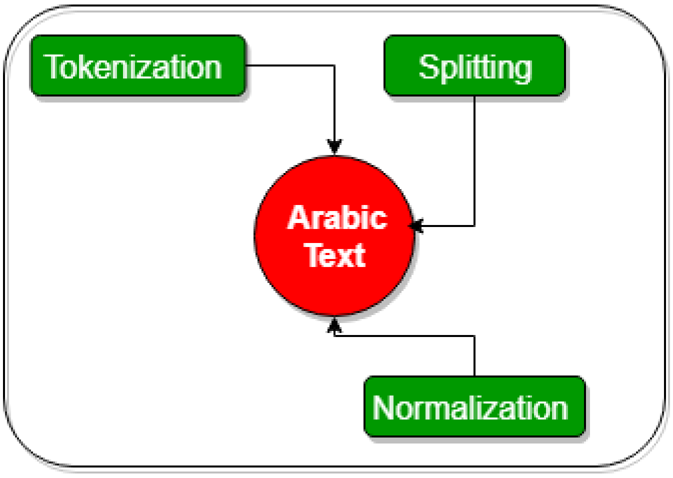


Figure (4.3): Pre-processing Stage

4.4 Global Terminologies

Global Terminologies are well known compound nouns on a global level in various fields such as medical, political and geographical…etc.

Such these terminologies will be collected as much as possible and classified in this thesis in order to contribute in the extraction of learning objectives by recognizing them in case they are mentioned in the Arabic text, In addition to classifying them into different fields, they will also be linked to the Arabic ontology tree because they eventually have an existential classification based on the origin of the term, For example, the word disease or "مرض" in Arabic, this word classified in the Arabic ontology tree being a single word but the disease "Aids" in Arabic "نقص المناعة المكتسبة" so because it's known globally it's called global and because it consist of three words instead of single word it's called terminology.

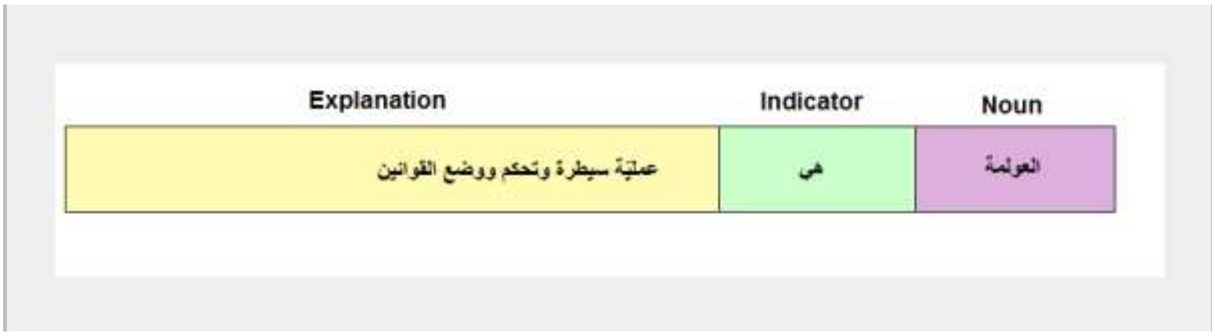
4.5 Templates for Learning Objectives

What is meant by templates is to unify the general form in which the learning objective comes in the Arabic text, as mentioned above, the first two levels of bloom's taxonomy (remembering and understanding) or also called (knowledge and comprehension), each of these two levels several learning objectives classifications, they are as follows:

4.5.1 Remembering level

Learning objectives that fall under this level are related to remembering and include the following classifications:

- <u>Definitions</u>: It is a definition and explanation of typical individual and compound nouns, therefore, the definition is a sentence consisting of a compound or individual noun followed by some indicators and then an explanation of that noun, so the general template for the definition is a statement contains noun, one of the definition indicators, they are as follows "means", "is", "meant by" in Arabic "هي" ,"هو", " بانها" ,"بانه","يعني","تعني" and the explanation as shown in the Figure (4, 5). And to extract the definitions we divide the text into sentences according to the sentences breaks we mentioned earlier, then we check each sentence if it matches the definition template or not which depending on degree of template matching (DTM) which is a degree indicates how much the sentence or text matches the learning objectives template, if the statement contains noun whether single or compound we increase the degree by one and if the statement contains indicators we increment the degree and so on, so higher degree means that this sentence is definitely a definition.



Explanation	Indicator	Noun
عملية سيطرة وتحكم ووضع القوانين	هي	العولمة

Figure (4.5): General Template for Definitions.

- Enumerations: Is a list of certain points whether sequentially or without sequence and these points may be attributes or characteristics, or branches or type…etc. These Enumerations also have a general template that is often comes similar to it and DTM is used to determine this similarity, and this generic template consists of a compound or individual noun and often contains a plural, a number whether these numbers are usual numbers or Arabic words and some indicators that indicate the enumerations like ("as follows", "following", "is") in Arabic ("الآتية", "يلي" ,"وهي" ,"التالية "), after that comes the enumeration points which are sentences that begin with a serial symbol or a symmetric symbol. And the method of extracting the enumerations as follows split the Arabic text to sentences and for each sentence check if this sentence contains one of enumeration indicators, so increment the DTM degree by one and if the sentence contains a number whether as numeric or as words or any word related to a number, also increase the DTM degree by one and if contains plural increment DTM again and if contains more than one plural word in this case we adopt only the first plural, so the degree becomes 3 which means the current statement is a statement for enumeration after that for each statement comes after the current statement directly and split it to Tokens and if begin with a serial symbol or a symmetric symbol, so this statement until the dot considered one of the enumeration points as shown in the Figure (4, 6).

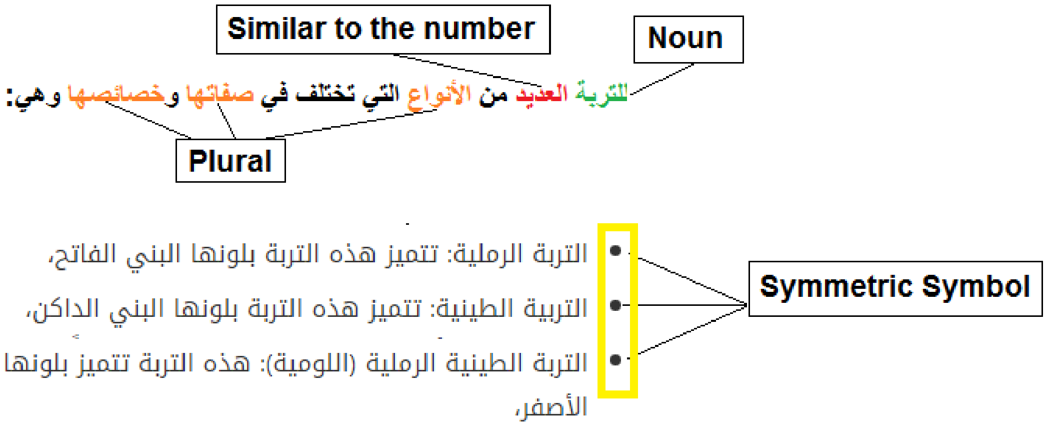

Similar to the number
Noun
للتربة العديد من الأنواع التي تختلف في صفاتها وخصائصها وهي:
Plural
التربة الرملية: تتميز هذه التربة بلونها البني الفاتح،
التربية الطينية: تتميز هذه التربة بلونها البني الداكن،
التربة الطينية الرملية (اللومية): هذه التربة تتميز بلونها الأصفر،
Symmetric Symbol

Figure (4.6): General Template for Enumerations

- Holy Quran: since the verses of Holy Quran are required to be remembered, especially in educational courses related to religious education, so extracting the verses of the Holy Quran and identifying them in the Arabic text be by retrieving all the verses of the Holy Quran in the Arabic text with the details of each verse, such as the verse number in the surah and the name and number of the surah.

4.5.2 Understanding level

Learning objectives that fall under this level are as close as they are to the understanding rather than remembering and include the following classifications:

- Comparisons: comparisons are distinctions and highlight the difference between two things, and these two things are single or composite nouns, therefore, we will note that its indicators in the sense of discrimination or difference, the indicators of this template in Arabic are ("الإختلاف بين" ,"الفرق بين"),

  Also in comparisons we will use the degree of template matching (DTM) to determine how closely the text matches the general template for comparisons, if this degree is high it means that we have a comparison between two nouns, the general template for the comparisons consists of one of the comparisons indicators and a pair of individual or compound nouns separated by one of the conjunction characters like "and", "و" in Arabic followed by a definition or explanation of each of the nouns mentioned by way of differentiation.

  In order to extract the comparisons from the Arabic text we must split the text firstly paragraph by paragraph then split each paragraph to sentences after that we check each of these sentences if the sentence contains one of the comparison indicators so we increment the DTM by one and if it contains pair of single or composite nouns separated by the conjunction character "و" in Arabic, we also increment DTM by one, and therefore if the degree is 2 this means this statement may be a comparison statement, but we cannot be sure yet, to make sure of that, we must check the remaining sentences in the current paragraph, If definitions or short explanations are found for each noun of the pair of nouns that we want to compare, It means that the current paragraph contains a comparison between these two nouns as shown in the Figure (4, 7)..

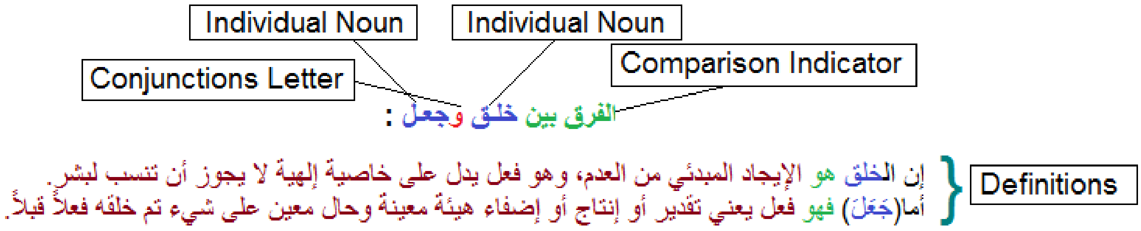

Individual Noun
Individual Noun
Conjunctions Letter
Comparison Indicator
الفرق بين خلق وجعل :
Definitions
إن الخلق هو الإيجاد المبدئي من العدم، وهو فعل يدل على خاصية إلهية لا يجوز أن تنسب لبشر.
أما(جَعَل) فهو فعل يعني تقدير أو إنتاج أو إضفاء هيئة معينة وحال معين على شيء تم خلقه فعلاً قبلاً.

Figure (4.7): General Template for Comparisons

- <u>Cause and Effect</u>: what is meant by cause and effect is a text contains both reason and result, and which makes us distinguish the section in the sentence that represents the cause and the section that represents the result is the existence of indicators for each of them:
  - The first case: the effect after the cause and the indicators are as follows: (“ أدى الى", "بسبب ذلك", "لذلك", "لذلك السبب", "نتيجة ل", "نتج عنه", "سبّب", "لهذا السبب").
  - The first case: the cause after the effectand the indicators are as follows: (“ سبب ذلك", "بسبب", "نتيجة ل", "سببه", "سببها", "ناتج عن", "ذلك بسبب").

In each of the above cases, we divide the text into sentences on the basis that the cause and effect are always mentioned in the same sentence and then check the sentences containing the cause and effect indicators and based on the type of indicators we identify the part that represents the cause and the part that represents the result.

4.6 Summary

In this chapter, we presented in details the approach which we will follow in this thesis in order to extract meaningful learning objectives, which is consist of four basic stages, these stages are: constructing Ontology tree, collecting and classifying data and preprocessing, collecting global terminology, and constructing general templates for learning objectives in the text, and we explained the details of each stage, including the definition of each stage and its sub-sections and how they were applied.

In the next chapter, we implement the approach to extract meaningful learning objectives.

Chapter 5

Implementation

Chapter 5

Implementation

In this chapter we will explain in details the implementation of the approach presented in the previous chapter for extracting learning objectives from Arabic text. First, we will explain the implementation environment including the programs, tools and systems used to implement the proposed model, then we will implement the stages that were detailed in the previous chapter one by one starting from Pre processing stage because it is considered the basis for the implementation of other stages, After that, we will be in the second stage, which is constructing the Arabic ontology tree, including tree design, methods of collecting tree words, and how to use them in analyzing Arabic text in order to extract learning objectives and obtain meaningful information through the use of this tree, After that we will discuss the implementation of the third phase, dealing with global terminology after clarifying what the global terminology and how to classify them in the previous chapter.

In this chapter we will explain the mechanism of implementation of this stage, including the method of collecting global terminology and how to use them to become an effective component and to contribute in extracting learning objectives from Arabic text, And finally the fourth stage is the implementation of the construction of general templates for learning objectives, also in the previous chapter we have explained what the general templates for the learning objectives and we drew the figures to illustrate them and what it is consist of and we explained how to match the text with these templates, but in this chapter we will implement what we explained earlier programmatically to extract learning objectives from Arabic Text.

This chapter is structured as follows: programs and tools and systems, implement preprocessing stage, ontology based extraction, extracting global terminologies, implement the general templates to extract learning objectives.

5.1 Programs and Tools

In order to implement the proposed approach, we used the following programs:

- SQL server and its attached tools like SQL server management studio, as a database environment to store and classify the data to be collected.

- Microsoft visual studio: it is an integrated development environment from Microsoft. It is used to develop computer programs, as well as web sites, web apps, web services and mobile apps (Wikipedia).
- Microsoft .NET Framework 4.5 with C# .NET Framework 4.5 is a highly compatible, in-place update to the .NET Framework 4. By using the .NET Framework 4.5 together with the C#, we can write Windows apps. The .NET Framework 4.5 includes significant language and framework enhancements for C#, (so that you can more easily write asynchronous code), the blending of control flow in synchronous code, a responsive UI, and web app scalability. The .NET Framework 4.5 adds substantial improvements to other functional areas such as ASP.NET, Managed Extensibility Framework, Windows Communication Foundation, Windows Workflow Foundation, and Windows Identity Foundation. The .NET Framework 4.5 delivers better performance, reliability, and security [16].

5.2 Implement Pre-processing Stage

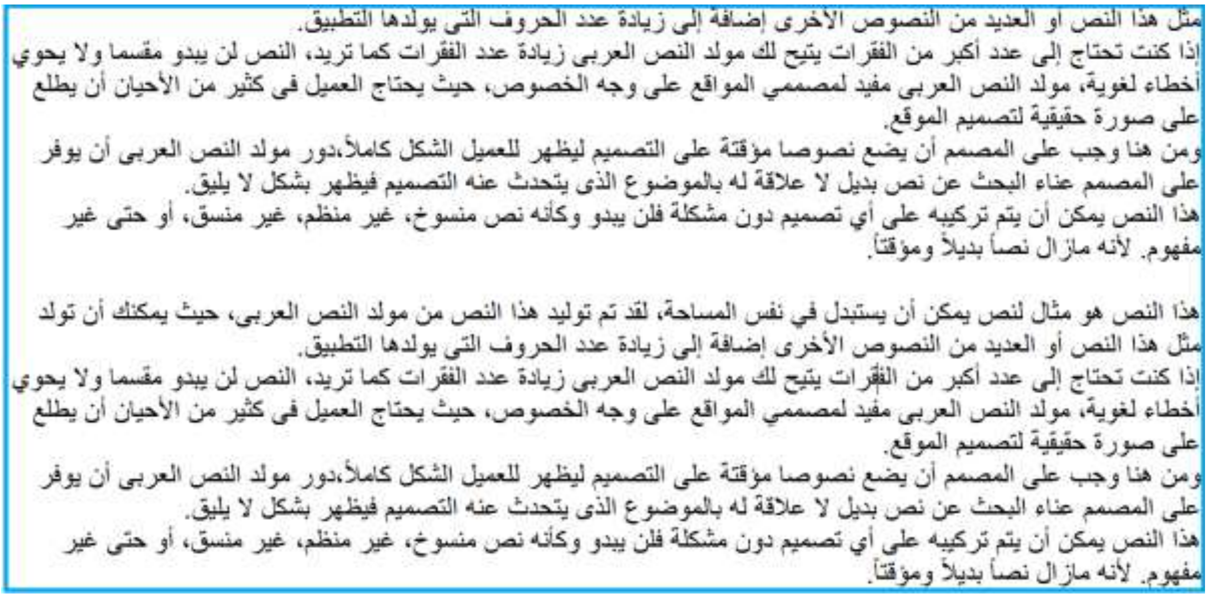
As we mentioned in the previous chapter, the pre-processing stage is a very important stage and all other stages depend on it. The aim of this stage is to prepare the text before and during the analysis of the text. It includes several sub-processes, these sub processes are splitting, normalization and nokenization after reading all text file that contains Arabic text, we will implement each sub process programmatically using a special function in C# .net language.

5.2.1 Splitting

Split the text in order to know its boundary, and the splitting process include split the text into paragraphs depending on paragraphs breaks

(>= Newline * 2) or split the text into statements depending on statements breaks (";", ".", ""”؟“ ,”؛, "?", "!", "،”) or split the text into words depending on possible words breaks (" ", "|", "\\", "/", ".", ";", ",", "؛", "?", "!", "(", ")", "،", "[", "]", "{", "}", "\"", "\'", "\t", "\n", "\r

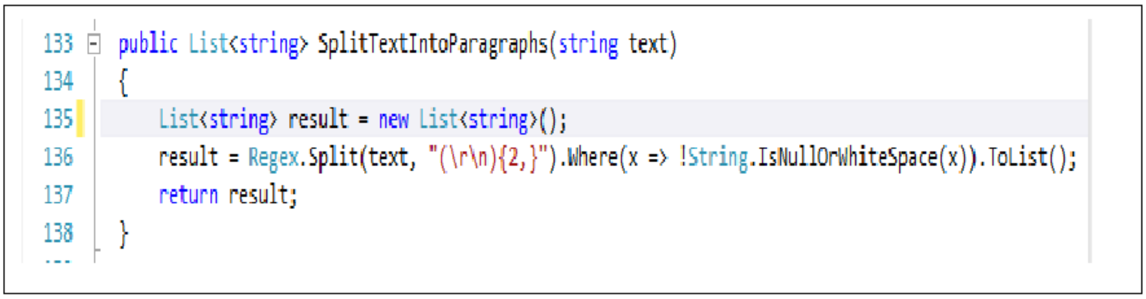
\n", "\n\n") this list of delimiters as well as a list of symbols that have been collected and categorized into the database together they form words delimiters, this splitting process will be performed by special C# code custom for each splitting method, the following code demonstrates how to perform paragraphs splitting, sentences splitting and words splitting as shown in Figure (5.1), Figure (5.2), Figure (5.3), Figure (5.4), Figure (5.5), Figure (5.6) and Figure (5.7).



مثل هذا النص أو العديد من النصوص الأخرى إضافة إلى زيادة عدد الحروف التى يولدها التطبيق.
إذا كنت تحتاج إلى عدد أكبر من الفقرات يتيح لك مولد النص العربى زيادة عدد الفقرات كما تريد، النص لن يبدو مقسما ولا يحوي
أخطاء لغوية، مولد النص العربى مفيد لمصممي المواقع على وجه الخصوص، حيث يحتاج العميل فى كثير من الأحيان أن يطلع
على صورة حقيقية لتصميم الموقع.
ومن هنا وجب على المصمم أن يضع نصوصا مؤقتة على التصميم ليظهر للعميل الشكل كاملاً،دور مولد النص العربى أن يوفر
على المصمم عناء البحث عن نص بديل لا علاقة له بالموضوع الذى يتحدث عنه التصميم فيظهر بشكل لا يليق.
هذا النص يمكن أن يتم تركيبه على أي تصميم دون مشكلة فلن يبدو وكأنه نص منسوخ، غير منظم، غير منسق، أو حتى غير
مفهوم. لأنه مازال نصاً بديلاً ومؤقتاً.

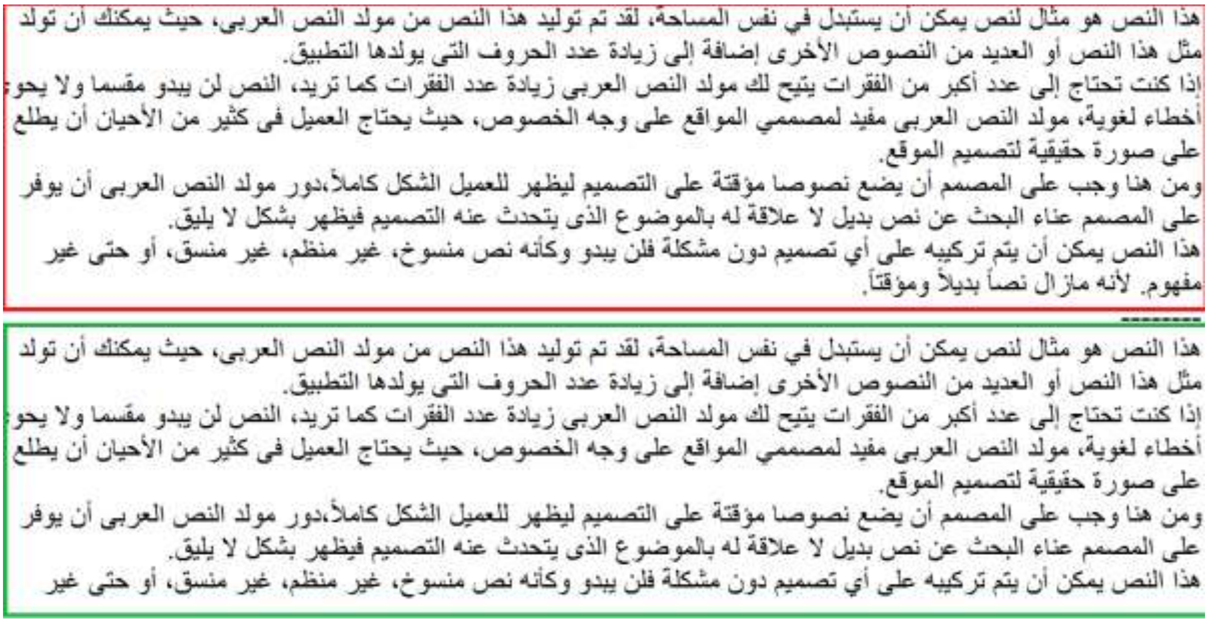
هذا النص هو مثال لنص يمكن أن يستبدل في نفس المساحة، لقد تم توليد هذا النص من مولد النص العربى، حيث يمكنك أن تولد
مثل هذا النص أو العديد من النصوص الأخرى إضافة إلى زيادة عدد الحروف التى يولدها التطبيق.
إذا كنت تحتاج إلى عدد أكبر من الفقرات يتيح لك مولد النص العربى زيادة عدد الفقرات كما تريد، النص لن يبدو مقسما ولا يحوي
أخطاء لغوية، مولد النص العربى مفيد لمصممي المواقع على وجه الخصوص، حيث يحتاج العميل فى كثير من الأحيان أن يطلع
على صورة حقيقية لتصميم الموقع.
ومن هنا وجب على المصمم أن يضع نصوصا مؤقتة على التصميم ليظهر للعميل الشكل كاملاً،دور مولد النص العربى أن يوفر
على المصمم عناء البحث عن نص بديل لا علاقة له بالموضوع الذى يتحدث عنه التصميم فيظهر بشكل لا يليق.
هذا النص يمكن أن يتم تركيبه على أي تصميم دون مشكلة فلن يبدو وكأنه نص منسوخ، غير منظم، غير منسق، أو حتى غير
مفهوم. لأنه مازال نصاً بديلاً ومؤقتاً.

Figure (5.1): Original Arabic text before splitting



```
public List<string> SplitTextIntoParagraphs(string text)
{
    List<string> result = new List<string>();
    result = Regex.Split(text, "(\r\n){2,}").Where(x => !String.IsNullOrWhiteSpace(x)).ToList();
    return result;
}
```

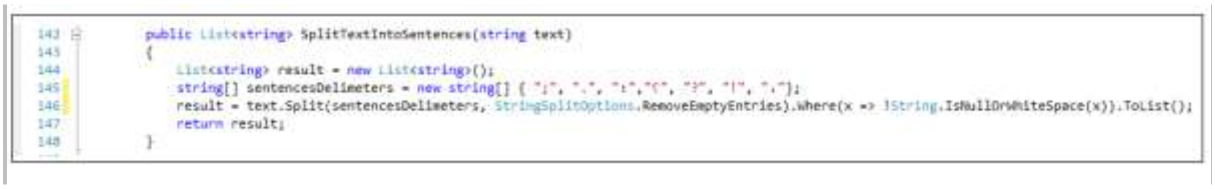
Figure (5.2): Code to Split Text into Paragraphs



هذا النص هو مثال لنص يمكن أن يستبدل في نفس المساحة، لقد تم توليد هذا النص من مولد النص العربى، حيث يمكنك أن تولد
مثل هذا النص أو العديد من النصوص الأخرى إضافة إلى زيادة عدد الحروف التى يولدها التطبيق.
إذا كنت تحتاج إلى عدد أكبر من الفقرات يتيح لك مولد النص العربى زيادة عدد الفقرات كما تريد، النص لن يبدو مقسما ولا يحو
أخطاء لغوية، مولد النص العربى مفيد لمصممي المواقع على وجه الخصوص، حيث يحتاج العميل فى كثير من الأحيان أن يطلع
على صورة حقيقية لتصميم الموقع.
ومن هنا وجب على المصمم أن يضع نصوصا مؤقتة على التصميم ليظهر للعميل الشكل كاملاً،دور مولد النص العربى أن يوفر
على المصمم عناء البحث عن نص بديل لا علاقة له بالموضوع الذى يتحدث عنه التصميم فيظهر بشكل لا يليق.
هذا النص يمكن أن يتم تركيبه على أي تصميم دون مشكلة فلن يبدو وكأنه نص منسوخ، غير منظم، غير منسق، أو حتى غير
مفهوم. لأنه مازال نصاً بديلاً ومؤقتاً.

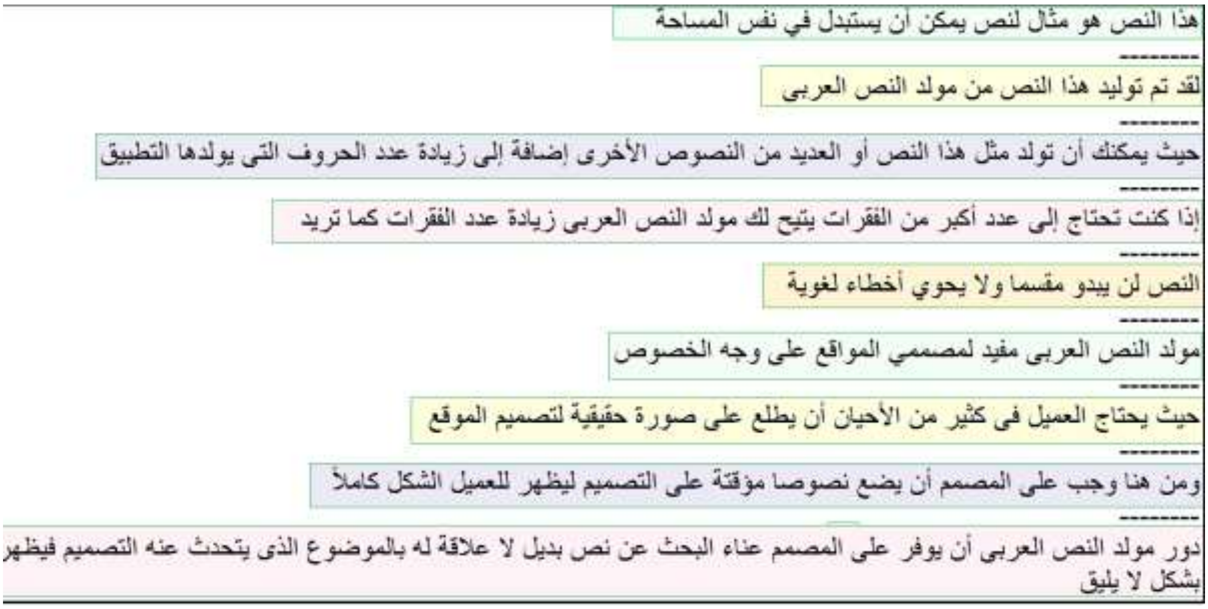
هذا النص هو مثال لنص يمكن أن يستبدل في نفس المساحة، لقد تم توليد هذا النص من مولد النص العربى، حيث يمكنك أن تولد
مثل هذا النص أو العديد من النصوص الأخرى إضافة إلى زيادة عدد الحروف التى يولدها التطبيق.
إذا كنت تحتاج إلى عدد أكبر من الفقرات يتيح لك مولد النص العربى زيادة عدد الفقرات كما تريد، النص لن يبدو مقسما ولا يحو
أخطاء لغوية، مولد النص العربى مفيد لمصممي المواقع على وجه الخصوص، حيث يحتاج العميل فى كثير من الأحيان أن يطلع
على صورة حقيقية لتصميم الموقع.
ومن هنا وجب على المصمم أن يضع نصوصا مؤقتة على التصميم ليظهر للعميل الشكل كاملاً،دور مولد النص العربى أن يوفر
على المصمم عناء البحث عن نص بديل لا علاقة له بالموضوع الذى يتحدث عنه التصميم فيظهر بشكل لا يليق.
هذا النص يمكن أن يتم تركيبه على أي تصميم دون مشكلة فلن يبدو وكأنه نص منسوخ، غير منظم، غير منسق، أو حتى غير

Figure (5.3): Split the Text into Paragraphs



```
        public List<string> SplitTextIntoSentences(string text)
        {
            List<string> result = new List<string>();
            string[] sentencesDelimeters = new string[] { "؛", ".", ":","،", "؟", "!", "." };
            result = text.Split(sentencesDelimeters, StringSplitOptions.RemoveEmptyEntries).Where(x => !String.IsNullOrWhiteSpace(x)).ToList();
            return result;
        }
```

Figure (5.4): Code to Split Text into Sentences.



هذا النص هو مثال لنص يمكن أن يستبدل في نفس المساحة

لقد تم توليد هذا النص من مولد النص العربى

حيث يمكنك أن تولد مثل هذا النص أو العديد من النصوص الأخرى إضافة إلى زيادة عدد الحروف التى يولدها التطبيق

إذا كنت تحتاج إلى عدد أكبر من الفقرات يتيح لك مولد النص العربى زيادة عدد الفقرات كما تريد

النص لن يبدو مقسما ولا يحوي أخطاء لغوية

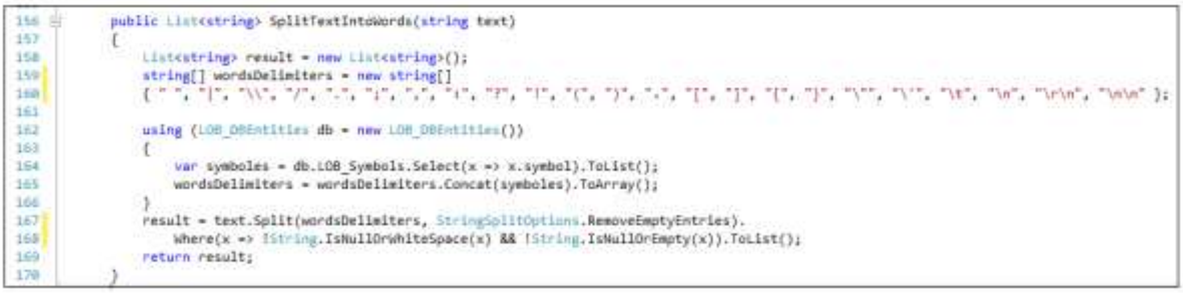
مولد النص العربى مفيد لمصممي المواقع على وجه الخصوص

حيث يحتاج العميل فى كثير من الأحيان أن يطلع على صورة حقيقية لتصميم الموقع

ومن هنا وجب على المصمم أن يضع نصوصا مؤقتة على التصميم ليظهر للعميل الشكل كاملاً

دور مولد النص العربى أن يوفر على المصمم عناء البحث عن نص بديل لا علاقة له بالموضوع الذى يتحدث عنه التصميم فيظهر بشكل لا يليق

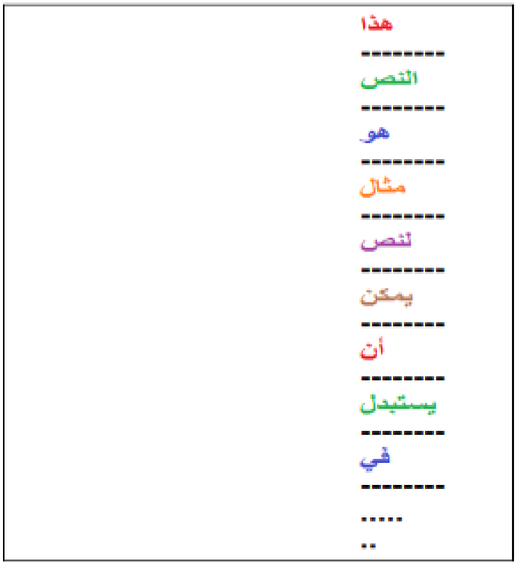
Figure (5.5): Split the Text into Sentences



```
public List<string> SplitTextIntoWords(string text)
{
    List<string> result = new List<string>();
    string[] wordsDelimiters = new string[]
    { " ", "|", "\\", "/", "-", ";", ",", ":", "?", "!", "(", ")", ".", "[", "]", "{", "}", "\"", "\'", "\t", "\n", "\r\n", "\n\n" };

    using (LOB_DBEntities db = new LOB_DBEntities())
    {
        var symboles = db.LOB_Symbols.Select(x => x.symbol).ToList();
        wordsDelimiters = wordsDelimiters.Concat(symboles).ToArray();
    }
    result = text.Split(wordsDelimiters, StringSplitOptions.RemoveEmptyEntries).
        Where(x => !String.IsNullOrWhiteSpace(x) && !String.IsNullOrEmpty(x)).ToList();
    return result;
}
```

Figure (5.6): Code to Split Text into Words.



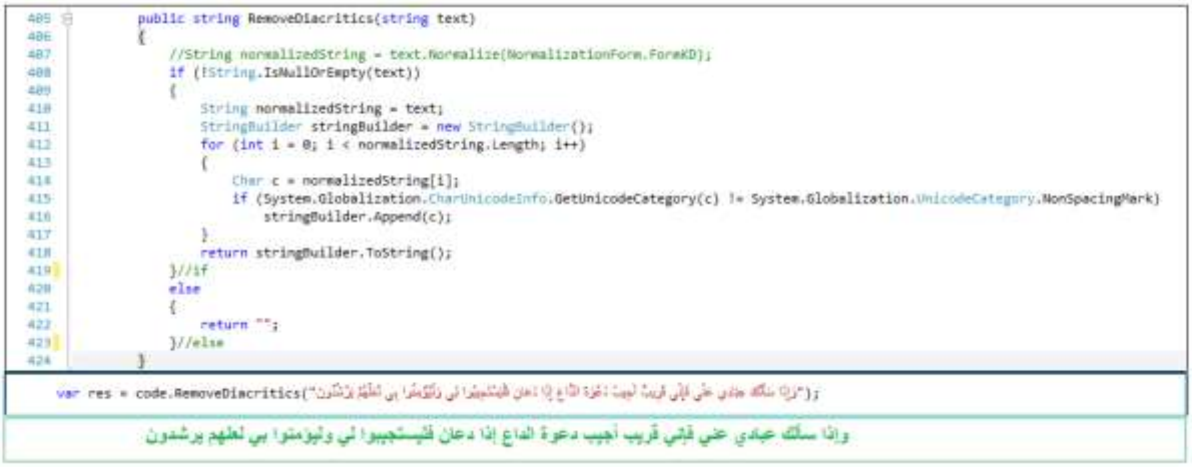
```
هذا
---------
النص
---------
هو
---------
مثال
---------
لنص
---------
يمكن
---------
أن
---------
يستبدل
---------
في
---------
-----
--
```

Figure (5.7): Split the Text into Words

5.2.2 Normalization

This process, as we have mentioned, involves several sub-processes such:

- Removing diacritics marks: it is function removes diacritics marks from the text as shown in Figure (5.8).



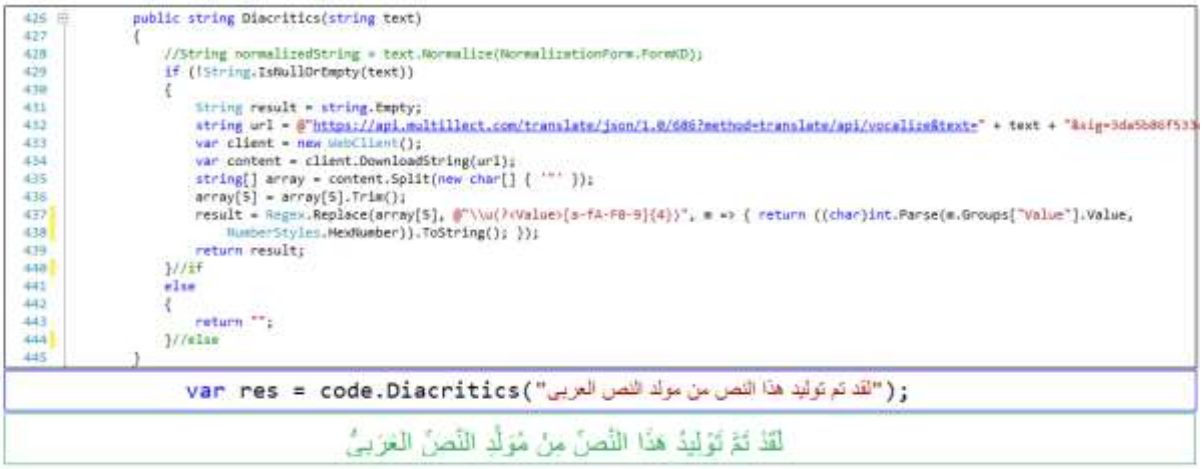
```
public string RemoveDiacritics(string text)
{
    //String normalizedString = text.Normalize(NormalizationForm.FormKD);
    if (!String.IsNullOrEmpty(text))
    {
        String normalizedString = text;
        StringBuilder stringBuilder = new StringBuilder();
        for (int i = 0; i < normalizedString.Length; i++)
        {
            Char c = normalizedString[i];
            if (System.Globalization.CharUnicodeInfo.GetUnicodeCategory(c) != System.Globalization.UnicodeCategory.NonSpacingMark)
                stringBuilder.Append(c);
        }
        return stringBuilder.ToString();
    }//if
    else
    {
        return "";
    }//else
}
```

```
var res = code.RemoveDiacritics("وَإِذَا سَأَلَكَ عِبَادِي عَنِّي فَإِنِّي قَرِيبٌ أُجِيبُ دَعْوَةَ الدَّاعِ إِذَا دَعَانِ فَلْيَسْتَجِيبُوا لِي وَلْيُؤْمِنُوا بِي لَعَلَّهُمْ يَرْشُدُونَ");
```

```
وإذا سألك عبادي عني فإني قريب أجيب دعوة الداع إذا دعان فليستجيبوا لي وليؤمنوا بي لعلهم يرشدون
```

Figure (5.8): Removing Diacritics Marks

- Placing diacritics marks: it is a function uses API for multillect.com website to assign diacritics marks for on the Arabic text as shown in Figure (5.9).



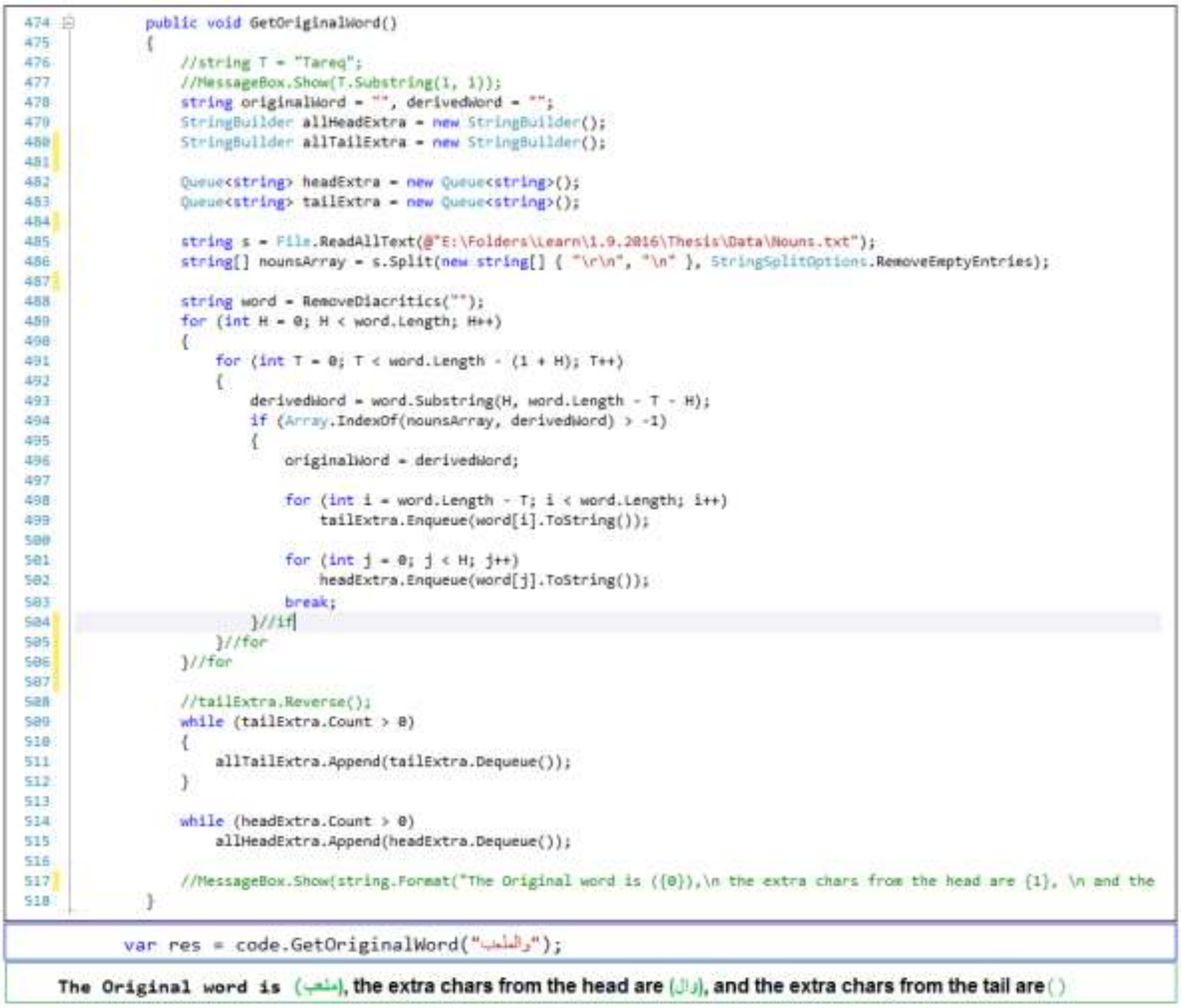
```
public string Diacritics(string text)
{
    //String normalizedString = text.Normalize(NormalizationForm.FormKD);
    if (!String.IsNullOrEmpty(text))
    {
        String result = string.Empty;
        string url = @"https://api.multillect.com/translate/json/1.0/686?method=translate/api/vocalize&text=" + text + "&sig=3da5b86f53
        var client = new WebClient();
        var content = client.DownloadString(url);
        string[] array = content.Split(new char[] { '"' });
        array[5] = array[5].Trim();
        result = Regex.Replace(array[5], @"\\u(?<Value>[a-fA-F0-9]{4})", m => { return ((char)int.Parse(m.Groups["Value"].Value,
            NumberStyles.HexNumber)).ToString(); });
        return result;
    }//if
    else
    {
        return "";
    }//else
}
```

```
var res = code.Diacritics("لقد تم توليد هذا النص من مولد النص العربي");
```

```
لَقَدْ تَمَّ تَوْلِيدُ هَذَا النَّصِّ مِنْ مُوَلِّدِ النَّصِّ العَرَبِيُّ
```

Figure (5.9): Placing Diacritics Marks

- Removing excess letters from the original word: it is a function take the word in the Arabic and works on extracting the original word from it and identifies the extra characters at the beginning and end of the word as shown in Figure (5.10).



```
public void GetOriginalWord()
{
    //string T = "Tareq";
    //MessageBox.Show(T.Substring(1, 1));
    string originalWord = "", derivedWord = "";
    StringBuilder allHeadExtra = new StringBuilder();
    StringBuilder allTailExtra = new StringBuilder();

    Queue<string> headExtra = new Queue<string>();
    Queue<string> tailExtra = new Queue<string>();

    string s = File.ReadAllText(@"E:\Folders\Learn\1.9.2016\Thesis\Data\Nouns.txt");
    string[] nounsArray = s.Split(new string[] { "\r\n", "\n" }, StringSplitOptions.RemoveEmptyEntries);

    string word = RemoveDiacritics("");
    for (int H = 0; H < word.Length; H++)
    {
        for (int T = 0; T < word.Length - (1 + H); T++)
        {
            derivedWord = word.Substring(H, word.Length - T - H);
            if (Array.IndexOf(nounsArray, derivedWord) > -1)
            {
                originalWord = derivedWord;

                for (int i = word.Length - T; i < word.Length; i++)
                    tailExtra.Enqueue(word[i].ToString());

                for (int j = 0; j < H; j++)
                    headExtra.Enqueue(word[j].ToString());
                break;
            }//if
        }//for
    }//for

    //tailExtra.Reverse();
    while (tailExtra.Count > 0)
    {
        allTailExtra.Append(tailExtra.Dequeue());
    }

    while (headExtra.Count > 0)
        allHeadExtra.Append(headExtra.Dequeue());

    //MessageBox.Show(string.Format("The Original word is ({0}),\n the extra chars from the head are {1}, \n and the
}
```

```
var res = code.GetOriginalWord("والملعب");
```

```
The Original word is (ملعب), the extra chars from the head are (وال), and the extra chars from the tail are ( )
```

Figure (5.10): Extract the Original Word

- Removing numbers, symbols and special characters from the word.

5.2.3 Tokenization

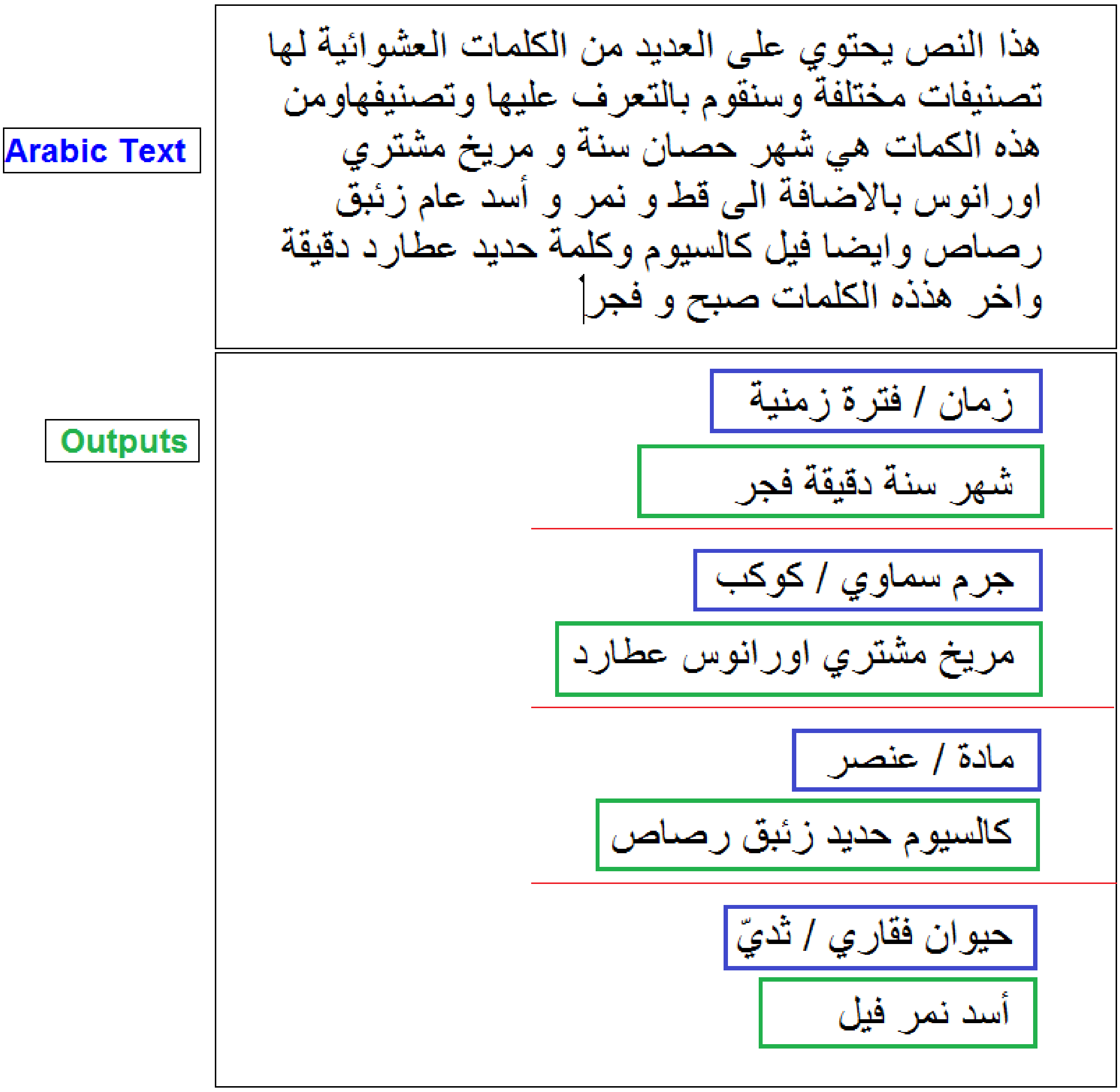
After reading the text file that contains Arabic text this process which is also a function written in C# .Net breaks down the text into segments called Tokens based on the white spaces among these segments and each of these Tokens may be either a word with its varieties or a number or symbol ... etc.

5.3 Ontology Based Extraction

After the design of the tree of the Arabic ontology, which includes a classification of the words of the Arabic language depending on the origin of the word, which began from the root word and branched to the sub-words and each word became represents a node in the tree has its own ID number and this ID carried by all the nouns that fall in the same classification, we also discussed previously that the way to construct the tree and assign the ID number to each node in the tree was manually by a table in the database.

Therefore, in this chapter we will explain the method of linking the nodes ID's in the tree with the collected nouns words, and this will be by collecting all the nouns of the same classification and the same origin in a text file, after that we reads all the words in this file and for each word we call a special function which take the word and the Node ID in the tree under which this word falls as function arguments, then this function inserts it into the nouns table with the ID number.

After enriching the tree with data and words, the question is how this tree can contribute to reinforcing the main research question by extracting meaningful learning objectives, the answer to this question depends on the accuracy of the classification the words in the ontology tree because the method of using the tree is considered static which will be by recognizing all the words in the Arabic text which have the same classification in the tree and listing them with their parent node word within the output as shown in Figure (5.11).



هذا النص يحتوي على العديد من الكلمات العشوائية لها
تصنيفات مختلفة وسنقوم بالتعرف عليها وتصنيفهاومن
هذه الكمات هي شهر حصان سنة و مريخ مشتري
اورانوس بالاضافة الى قط و نمر و أسد عام زئبق
رصاص وايضا فيل كالسيوم وكلمة حديد عطارد دقيقة
واخر هذذه الكلمات صبح و فجر

زمان / فترة زمنية

شهر سنة دقيقة فجر

جرم سماوي / كوكب

مريخ مشتري اورانوس عطارد

مادة / عنصر

كالسيوم حديد زئبق رصاص

حيوان فقاري / ثديّ

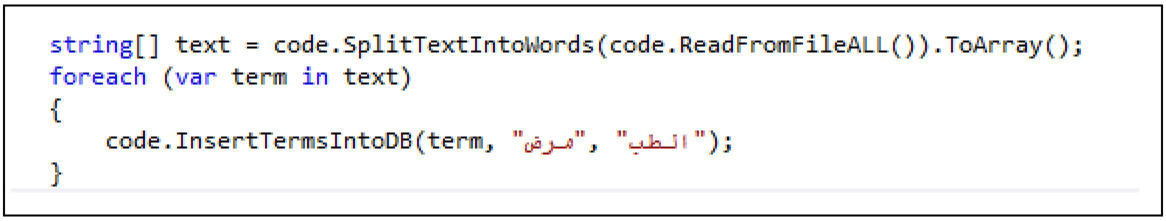
أسد نمر فيل

Figure (5.11): Ontology Based Extraction

5.4 Extracting Global Terminologies

Global Terminologies are compound nouns that universally recognized and cannot be included and categorized in the ontology tree because it is compound nouns and not individual words, after explaining the method of classification of these global terms previously, we should discuss in this chapter how to collect this data and how to exploit and use it to enrich the main goal for this research.

And regarding the methods of collecting global terms is to collect all the terms from the same field and the same classification in a text file and then by a specialized code that reads the whole file and for each global term in the text file we call specialized function which take the term and the domain to which the term belongs and key word for the term which is considered the real classification for the term, through which it can be linked to the tree of ontology and inserts them into a table in the database as shown in Figure(5.12).



```
string[] text = code.SplitTextIntoWords(code.ReadFromFileALL()).ToArray();
foreach (var term in text)
{
    code.InsertTermsIntoDB(term, "مرض", "الطب");
}
```

Figure (5.12): Code to Collect Global Terminology

As for the contribution of these terms to extract useful information from the Arabic text, simply by identifying all the global terms found in the text with their own classification and trying to link them to the tree of the ontology to know also the keyword classification of the terms in the ontology tree.

5.5 Implement the General Templates to Extract LO

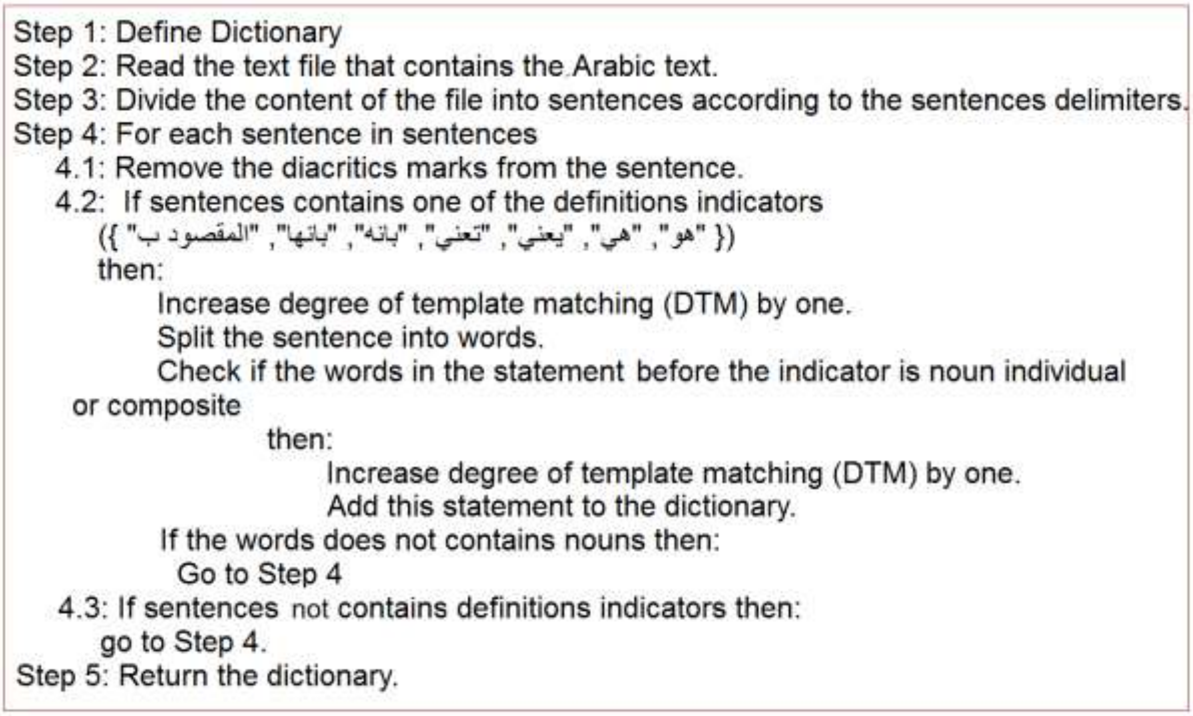
After we talked about grouping learning objectives under the first two levels of the Blooms Taxonomy in the previous chapter and we have discussed each level alone and what are the learning objectives that fall under each level, and then we have built a general template for each group of learning objectives and we explained the method of matching the text with the general templates to extract learning objectives, and now in this chapter we will apply the construction of general templates and will explain in more details how to match them with the Arabic text until we extract meaningful learning objectives.

5.5.1 Remembering level

Groups of learning objectives falling under this level as we mentioned earlier are:

- <u>Definitions</u>: the method of extracting definitions from the Arabic text in as the following:

First, we read the text file that contains the◌ِ Arabic text and divide the content of the file into sentences according to the sentences delimiters, then for each sentence in sentences we remove the diacritics marks from the sentence, and check If the sentence contains one of the definitions indicators then we increase degree of template matching (DTM) by one, and split the sentence into words, after that we check if the words in the statement before the indicator is noun whether is individual or composite or not, if it's noun so we increase degree of DTM by one, and add this statement to the result dictionary, finally the function return the result dictionary as shown in Figure (5.13).

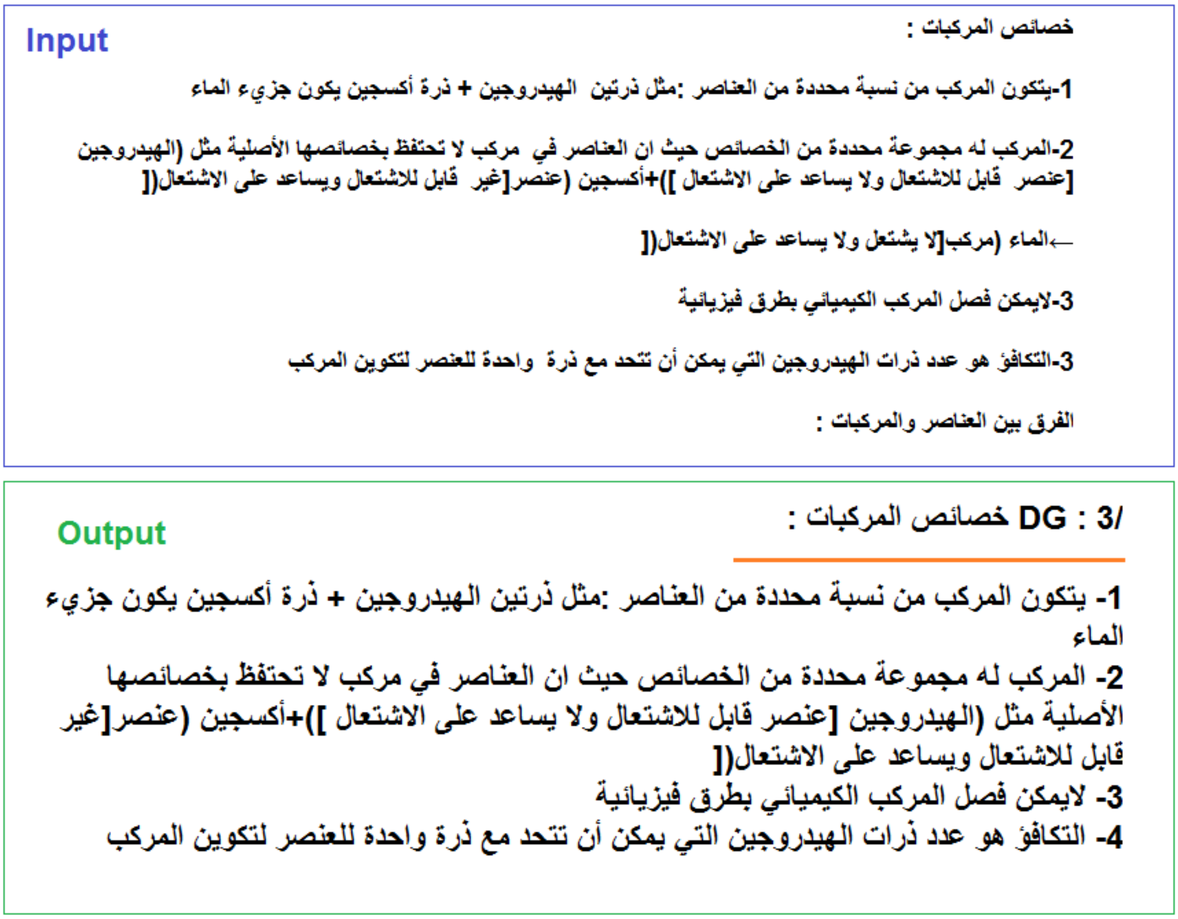


```
Step 1: Define Dictionary
Step 2: Read the text file that contains the Arabic text.
Step 3: Divide the content of the file into sentences according to the sentences delimiters.
Step 4: For each sentence in sentences
    4.1: Remove the diacritics marks from the sentence.
    4.2:  If sentences contains one of the definitions indicators
        ({"هو", "هي", "يعني", "تعني", "بانه", "بانها", "المقصود ب"})
        then:
            Increase degree of template matching (DTM) by one.
            Split the sentence into words.
            Check if the words in the statement before the indicator is noun individual
   or composite
                    then:
                        Increase degree of template matching (DTM) by one.
                        Add this statement to the dictionary.
            If the words does not contains nouns then:
              Go to Step 4
    4.3: If sentences not contains definitions indicators then:
        go to Step 4.
Step 5: Return the dictionary.
```

Figure (5.13): Algorithm to Extract Definitions

- <u>Enumerations</u>: the method of extracting Enumerations from the Arabic text in as the following:

  First, we read the text file that contains the◌ِ Arabic text and divide the content of the file into sentences according to the sentences delimiters, then for each sentence in sentences we remove the diacritics marks from the sentence, and check If the sentence contains one of the Enumerations indicators then we increase DTM by one, and if the sentence contains nouns also we increase DTM by one, and if the sentence contains plural increase DTM by one again then check all the next sentences and return any statement begin with sequential or symmetric symbol, after that we add the enumeration statement and its sub-statement (enumeration points) to the result dictionary finally the function return the result dictionary as shown in the Figure (5.14).



خصائص المركبات :

1-يتكون المركب من نسبة محددة من العناصر :مثل ذرتين الهيدروجين + ذرة أكسجين يكون جزيء الماء

2-المركب له مجموعة محددة من الخصائص حيث ان العناصر في مركب لا تحتفظ بخصائصها الأصلية مثل (الهيدروجين
[عنصر قابل للاشتعال ولا يساعد على الاشتعال])+أكسجين (عنصر[غير قابل للاشتعال ويساعد على الاشتعال)]

←الماء (مركب[لا يشتعل ولا يساعد على الاشتعال)]

3-لايمكن فصل المركب الكيميائي بطرق فيزيائية

3-التكافؤ هو عدد ذرات الهيدروجين التي يمكن أن تتحد مع ذرة واحدة للعنصر لتكوين المركب

الفرق بين العناصر والمركبات :

3/ : DG خصائص المركبات :

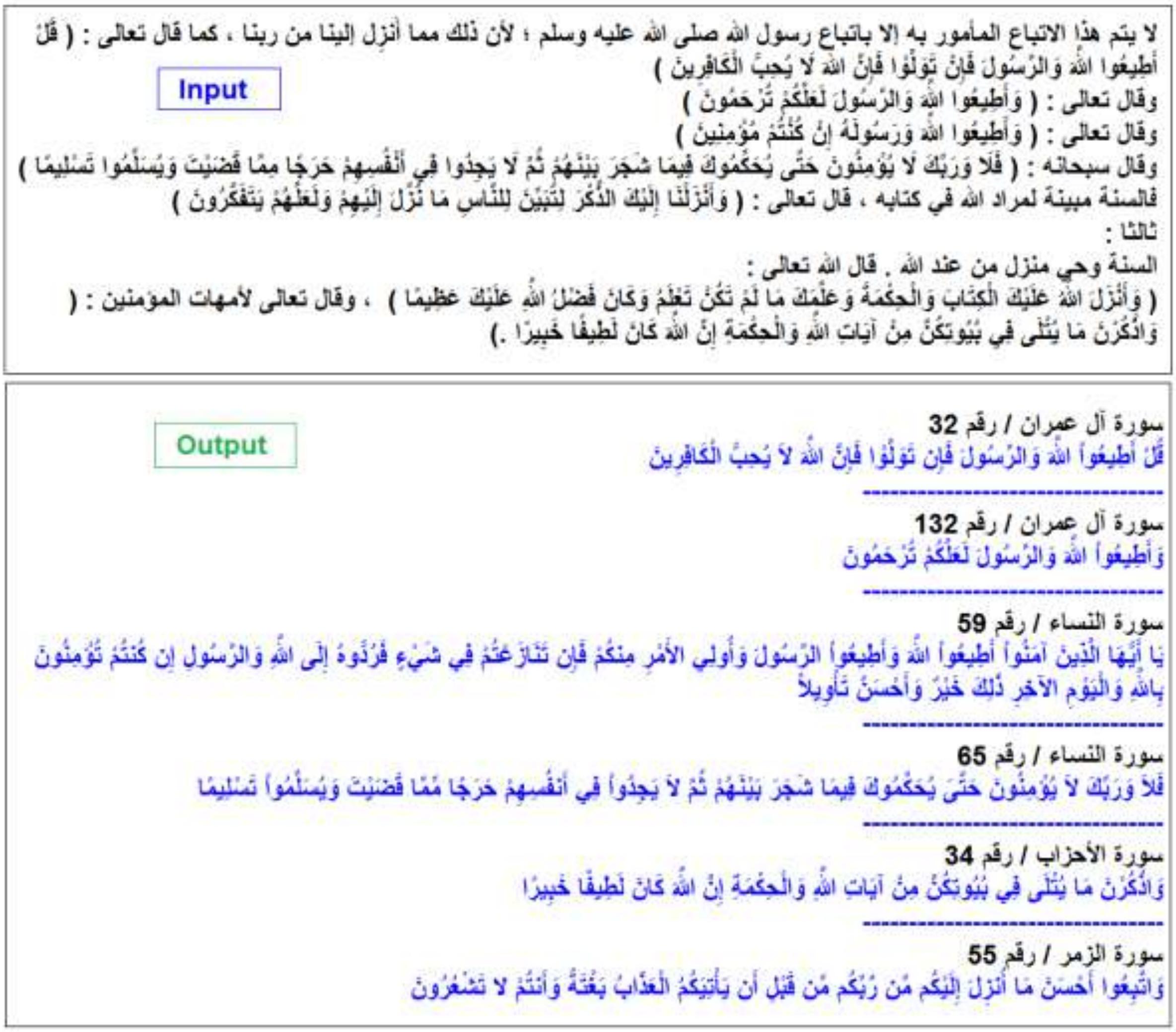
1- يتكون المركب من نسبة محددة من العناصر :مثل ذرتين الهيدروجين + ذرة أكسجين يكون جزيء
الماء
2- المركب له مجموعة محددة من الخصائص حيث ان العناصر في مركب لا تحتفظ بخصائصها
الأصلية مثل (الهيدروجين [عنصر قابل للاشتعال ولا يساعد على الاشتعال])+أكسجين (عنصر[غير
قابل للاشتعال ويساعد على الاشتعال)]
3- لايمكن فصل المركب الكيميائي بطرق فيزيائية
4- التكافؤ هو عدد ذرات الهيدروجين التي يمكن أن تتحد مع ذرة واحدة للعنصر لتكوين المركب

Figure (5.14): Extracting the Enumerations.

- <u>Verses of the Holy Quran</u>

Extracting verses from the Holy Quran from the Arabic text is very important especially in the courses related to religion, since the recitation of the Quranic verses is classified under the first level of the Bloom's Taxonomy. Therefore, the verses of the Holy Quran were fully collected and classified according to the name of the surah and the number of the verse and kept in the database. Although the collection of the verses was very difficult because of the scarcity of sources containing the entire Holy Quran in the form of texts, but the collection process was very accurate and free of any language errors.

And regarding implementing extraction of the Holy Quran verses from the Arabic text is by dividing the text into sentences, taking into account that sentences delimiters in this case more comprehensive to include all the symbols well known, after that any sentence that does not contain a holy Quran verse is excluded, and for each sentence from the remaining sentences diacritical marks removed from this statement because these diacritical marks may differ on the Quranic verses according to the different sources. Hence, the details of this verse are retrieved from the database such as the name of the surah in which this verse exist and the number of this verse in the surah, and the results obtained are as shown in Figure (5.15).



لا يتم هذا الاتباع المأمور به إلا باتباع رسول الله صلى الله عليه وسلم ؛ لأن ذلك مما أنزل إلينا من ربنا ، كما قال تعالى : (قُلْ أَطِيعُوا اللهَ وَالرَّسُولَ فَإِنْ تَوَلَّوْا فَإِنَّ اللهَ لَا يُحِبُّ الْكَافِرِينَ)
وقال تعالى : (وَأَطِيعُوا اللهَ وَالرَّسُولَ لَعَلَّكُمْ تُرْحَمُونَ)
وقال تعالى : (وَأَطِيعُوا اللهَ وَرَسُولَهُ إِنْ كُنْتُمْ مُؤْمِنِينَ)
وقال سبحانه : (فَلَا وَرَبِّكَ لَا يُؤْمِنُونَ حَتَّى يُحَكِّمُوكَ فِيمَا شَجَرَ بَيْنَهُمْ ثُمَّ لَا يَجِدُوا فِي أَنْفُسِهِمْ حَرَجًا مِمَّا قَضَيْتَ وَيُسَلِّمُوا تَسْلِيمًا)
فالسنة مبينة لمراد الله في كتابه ، قال تعالى : (وَأَنْزَلْنَا إِلَيْكَ الذِّكْرَ لِتُبَيِّنَ لِلنَّاسِ مَا نُزِّلَ إِلَيْهِمْ وَلَعَلَّهُمْ يَتَفَكَّرُونَ)
ثالثا :
السنة وحي منزل من عند الله . قال الله تعالى :
(وَأَنْزَلَ اللهُ عَلَيْكَ الْكِتَابَ وَالْحِكْمَةَ وَعَلَّمَكَ مَا لَمْ تَكُنْ تَعْلَمُ وَكَانَ فَضْلُ اللهِ عَلَيْكَ عَظِيمًا) ، وقال تعالى لأمهات المؤمنين : (وَاذْكُرْنَ مَا يُتْلَى فِي بُيُوتِكُنَّ مِنْ آيَاتِ اللهِ وَالْحِكْمَةِ إِنَّ اللهَ كَانَ لَطِيفًا خَبِيرًا .)

سورة آل عمران / رقم 32
قُلْ أَطِيعُواْ اللّهَ وَالرَّسُولَ فإِن تَوَلَّوْاْ فَإِنَّ اللّهَ لاَ يُحِبُّ الْكَافِرِينَ

سورة آل عمران / رقم 132
وَأَطِيعُواْ اللّهَ وَالرَّسُولَ لَعَلَّكُمْ تُرْحَمُونَ

سورة النساء / رقم 59
يَا أَيُّهَا الَّذِينَ آمَنُواْ أَطِيعُواْ اللّهَ وَأَطِيعُواْ الرَّسُولَ وَأُوْلِي الأَمْرِ مِنكُمْ فَإِن تَنَازَعْتُمْ فِي شَيْءٍ فَرُدُّوهُ إِلَى اللّهِ وَالرَّسُولِ إِن كُنتُمْ تُؤْمِنُونَ بِاللّهِ وَالْيَوْمِ الآخِرِ ذَلِكَ خَيْرٌ وَأَحْسَنُ تَأْوِيلاً

سورة النساء / رقم 65
فَلاَ وَرَبِّكَ لاَ يُؤْمِنُونَ حَتَّىَ يُحَكِّمُوكَ فِيمَا شَجَرَ بَيْنَهُمْ ثُمَّ لاَ يَجِدُواْ فِي أَنفُسِهِمْ حَرَجًا مِّمَّا قَضَيْتَ وَيُسَلِّمُواْ تَسْلِيمًا

سورة الأحزاب / رقم 34
وَاذْكُرْنَ مَا يُتْلَى فِي بُيُوتِكُنَّ مِنْ آيَاتِ اللَّهِ وَالْحِكْمَةِ إِنَّ اللَّهَ كَانَ لَطِيفًا خَبِيرًا

سورة الزمر / رقم 55
وَاتَّبِعُوا أَحْسَنَ مَا أُنزِلَ إِلَيْكُم مِّن رَّبِّكُم مِّن قَبْلِ أَن يَأْتِيَكُمُ الْعَذَابُ بَغْتَةً وَأَنتُمْ لَا تَشْعُرُونَ

Figure (5.15): Extracting Verses of the Holy Quran

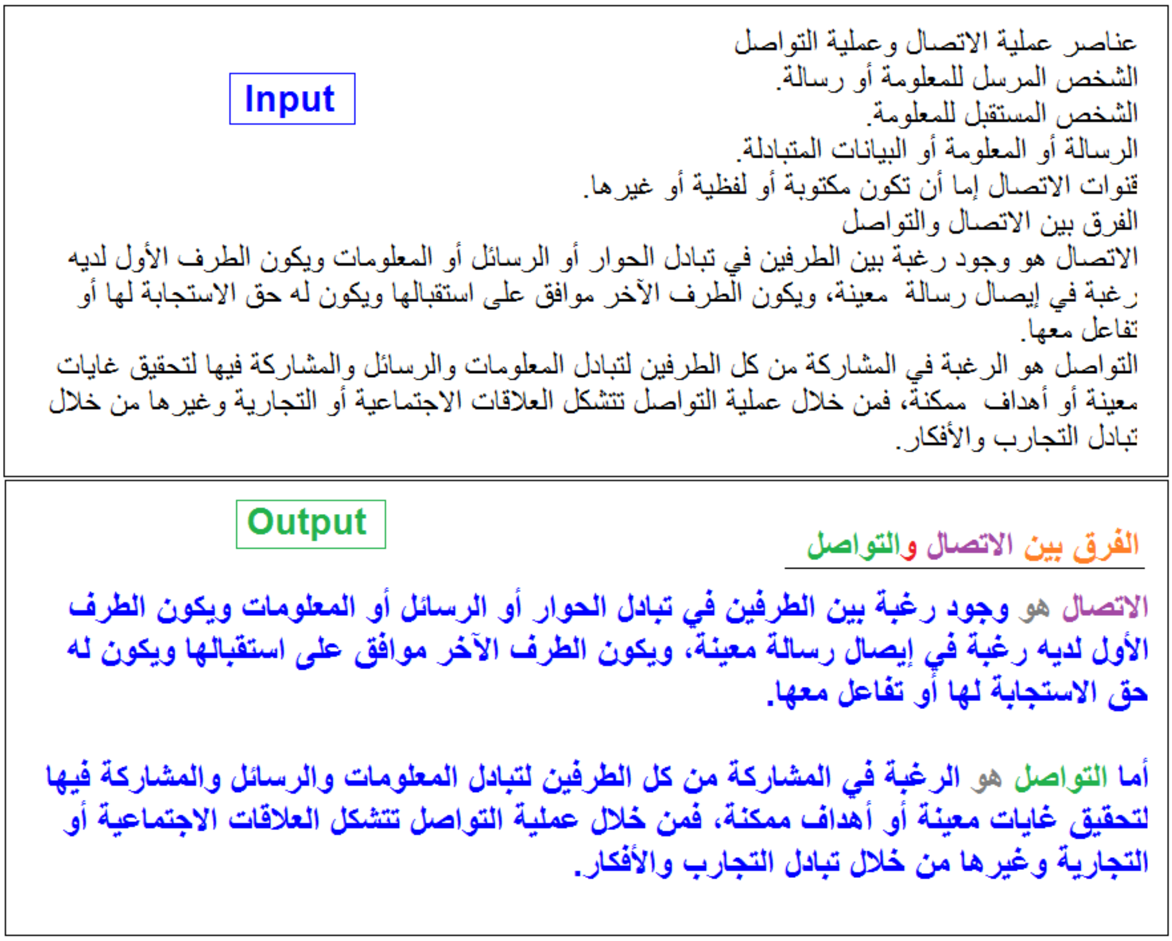
5.5.2 Understanding level

This level includes groups of learning objectives closest to understanding them than to remember and memorize them, which includes the following groups:

- Comparisons: regarding the comparisons we have talked about it in the previous chapter very much and in detail where we explained what is meant by comparisons and what is the general template for the comparisons and what are the most famous indicators, and also we have explained how to extract the comparisons from the Arabic text programmatically depending on the existence of its indicators and the high degree of template matching (DTM). But in this chapter we will not add much to what we explained previously, except for a detailed description of how to implement the extraction of comparisons, the general template for the comparisons as follows:

  The comparison statement which includes: one of comparisons indicators which are: ("الفرق بين", "الاختلاف بين")., the first term which may be a single or a compound noun, conjunction characters like "and", "و" in Arabic and the second term also may be a single or a compound noun.

  Therefore, the method of implementation in more details by reading the text file that contains the Arabic text and splitting it into sentences as usual, and for each sentence we remove the diacritical marks and then examine if it contains one of the comparisons indicators, if contains so split he sentence into list of words and get the index of the indicator from this list then all the words in the list from indicator index to the conjunction character will form together the first term and all the words in the list from the index of conjunction character "و" to the end of statement will form together the second term, after we know the first and the second term which we want to compare them, then we search for a definition or any explanation for each of them in the sentences following the current sentence (comparison statement) of the same paragraph If we find an explanation for each of them, all we do as a last step is to add both the comparison statement and the terms definitions to the dictionary as a final result and return the dictionary as in Figure (5.16).



Input

عناصر عملية الاتصال وعملية التواصل
الشخص المرسل للمعلومة أو رسالة.
الشخص المستقبل للمعلومة.
الرسالة أو المعلومة أو البيانات المتبادلة.
قنوات الاتصال إما أن تكون مكتوبة أو لفظية أو غيرها.
الفرق بين الاتصال والتواصل
الاتصال هو وجود رغبة بين الطرفين في تبادل الحوار أو الرسائل أو المعلومات ويكون الطرف الأول لديه رغبة في إيصال رسالة معينة، ويكون الطرف الآخر موافق على استقبالها ويكون له حق الاستجابة لها أو تفاعل معها.
التواصل هو الرغبة في المشاركة من كل الطرفين لتبادل المعلومات والرسائل والمشاركة فيها لتحقيق غايات معينة أو أهداف ممكنة، فمن خلال عملية التواصل تتشكل العلاقات الاجتماعية أو التجارية وغيرها من خلال تبادل التجارب والأفكار.

Output

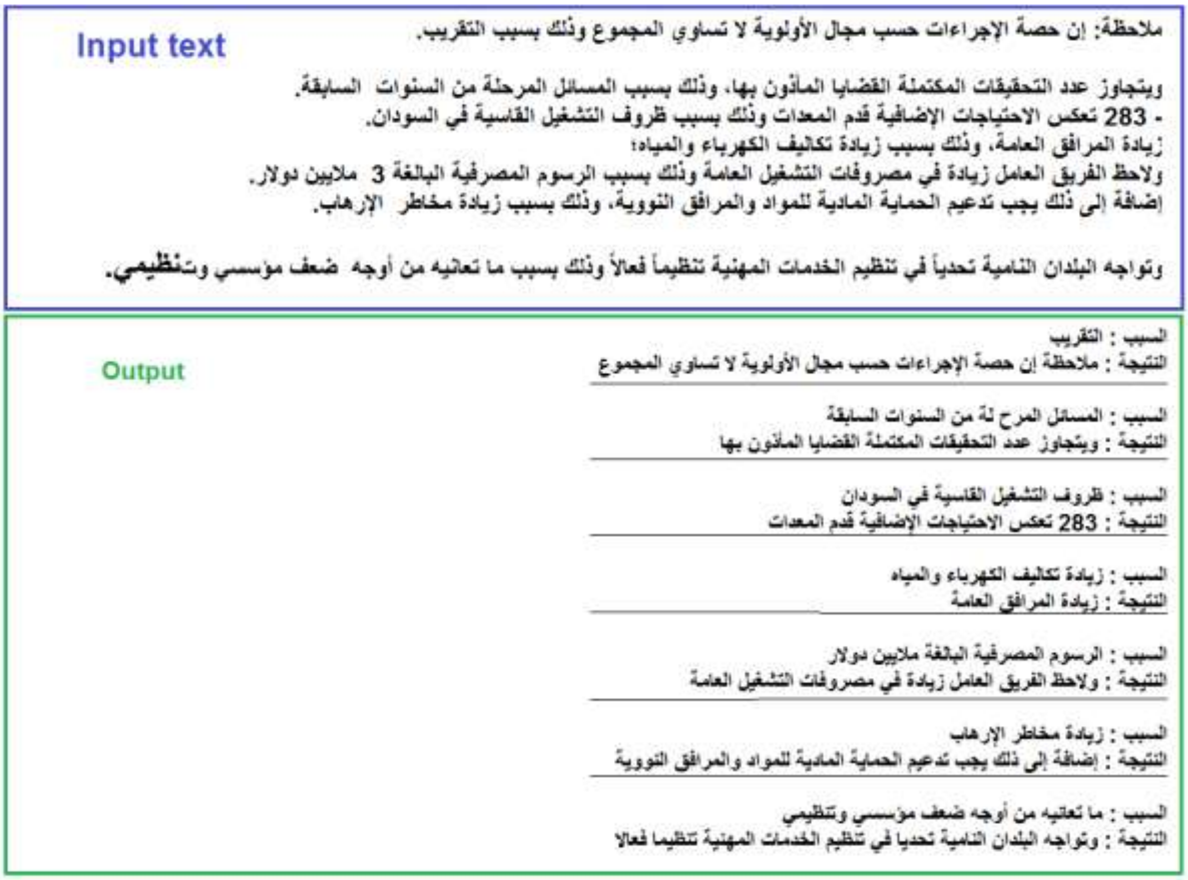
الفرق بين الاتصال والتواصل

الاتصال هو وجود رغبة بين الطرفين في تبادل الحوار أو الرسائل أو المعلومات ويكون الطرف الأول لديه رغبة في إيصال رسالة معينة، ويكون الطرف الآخر موافق على استقبالها ويكون له حق الاستجابة لها أو تفاعل معها.

أما التواصل هو الرغبة في المشاركة من كل الطرفين لتبادل المعلومات والرسائل والمشاركة فيها لتحقيق غايات معينة أو أهداف ممكنة، فمن خلال عملية التواصل تتشكل العلاقات الاجتماعية أو التجارية وغيرها من خلال تبادل التجارب والأفكار.

Figure (5.16): Extracting Comparisons

- <u>Cause and Effect</u>: The method of extracting the cause and the result in detail by dividing the text into sentences and for each sentence the diacritical marks are removed then we check if this sentence contains the indicators of the cause after the effect or the indicators of the effect after the cause and then we divide the sentence into two parts and finally we remove the single letters from each part.



Input text

ملاحظة: إن حصة الإجراءات حسب مجال الأولوية لا تساوي المجموع وذلك بسبب التقريب.

ويتجاوز عدد التحقيقات المكتملة القضايا المأذون بها، وذلك بسبب المسائل المرحلة من السنوات السابقة.
- 283 تعكس الاحتياجات الإضافية قدم المعدات وذلك بسبب ظروف التشغيل القاسية في السودان.
زيادة المرافق العامة، وذلك بسبب زيادة تكاليف الكهرباء والمياه؛
ولاحظ الفريق العامل زيادة في مصروفات التشغيل العامة وذلك بسبب الرسوم المصرفية البالغة 3 ملايين دولار.
إضافة إلى ذلك يجب تدعيم الحماية المادية للمواد والمرافق النووية، وذلك بسبب زيادة مخاطر الإرهاب.

وتواجه البلدان النامية تحدياً في تنظيم الخدمات المهنية تنظيماً فعالاً وذلك بسبب ما تعانيه من أوجه ضعف مؤسسي وتنظيمي.

Output

السبب : التقريب
النتيجة : ملاحظة إن حصة الإجراءات حسب مجال الأولوية لا تساوي المجموع

السبب : المسائل المرح لة من السنوات السابقة
النتيجة : ويتجاوز عدد التحقيقات المكتملة القضايا المأذون بها

السبب : ظروف التشغيل القاسية في السودان
النتيجة : 283 تعكس الاحتياجات الإضافية قدم المعدات

السبب : زيادة تكاليف الكهرباء والمياه
النتيجة : زيادة المرافق العامة

السبب : الرسوم المصرفية البالغة ملايين دولار
النتيجة : ولاحظ الفريق العامل زيادة في مصروفات التشغيل العامة

السبب : زيادة مخاطر الإرهاب
النتيجة : إضافة إلى ذلك يجب تدعيم الحماية المادية للمواد والمرافق النووية

السبب : ما تعانيه من أوجه ضعف مؤسسي وتنظيمي
النتيجة : وتواجه البلدان النامية تحديا في تنظيم الخدمات المهنية تنظيما فعالا

Figure (5.17): Extracting Cause-Effect

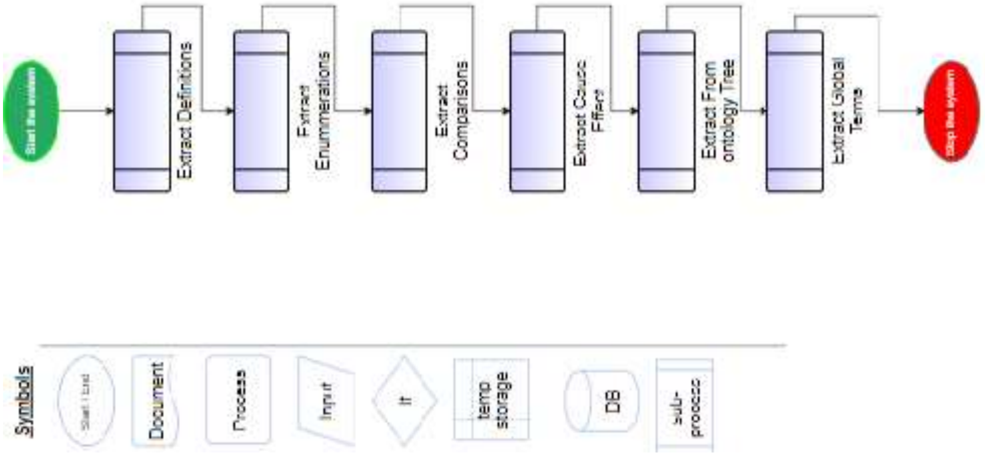


Figure (5.18): Implementation Flowchart

5.6 Summary

In this chapter, we presented details about the implementation of our approach, including the tools and programs required for the implementation, and then the stages of implementation starting from the pre-processing stage, the stage of extracting the words from the tree of ontology, the extraction of global terminology and finally the extraction of learning objectives based on predefined general templates, including an explanation of the implementation method in details for each stage. In addition, a sample of the extracted results was documented at each stage, illustrating the input text and the extracted result.

In the next chapter, we will analyze 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, we present the experimental results for a large and formal Arabic text such as a curriculum for school students on different subjects, and analyzing the results to prove that the proposed approach can exploit the Ontology tree to identify all the entities in the Arabic text and to classify them according to their origins and also to identify the common global terminology, and to prove also the approach is able to extract meaningful learning objectives based on the general templates built in advance by matching the text with these templates , In addition, evaluate the performance of the proposed approach in extracting the learning objectives, extracting the Quranic verses and identifying the basic terminologies in the Arabic text, and finally, visualize the obtained results using the graphical representation to give a formal view for the results and compare them.

This chapter is structured as follows: experiments setup, preparation**,** Arabic text files and scenarios**,** evaluation of the approach and discussion.

6.1 Experiments Setup

To perform the experiment and implement the proposed approach, we need to use many programs, tools and programming languages within the experiment environment in which the experiment will be conducted. We have talked about these programs briefly in the previous chapter (Chapter 5), but now we will describe them in some detail.

- SQL server, SQL server management studio, used to build a database, tables that will retain data such as nouns, verbs, parts of speech, numbers, symbols, and structure of an ontology tree.

- Microsoft visual studio: Microsoft Visual Studio is a (IDE), used as a development environment to write code in C# language with .NET Framework Platform. Visual

Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silver light. It can produce both native code and managed code, also includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring (Wikipedia).

- C# Language and the .NET Framework: C# Language is highly expressive, simple and easy to learn and it's also an object-oriented language supports the concepts of encapsulation, inheritance, and polymorphism. C# programs run on the .NET Framework, an integral component of Windows that includes a virtual execution system called the common language runtime (CLR) and a unified set of class libraries [17].
- Notepad: notepad is a simple text editor for Microsoft Windows (Wikipedia), It is used as a source of Arabic text to be entered and then analyzed.
- Operating System: the description of operating system and the computer that are considered the platform for conducting the experiment as shown in the Table (5.1).

Table (6.1): Computer Specifications

Operating system	**Windows 7**
System type	64-bit Operating System
Processor	Intel(R) Core(TM) i5 -2410M CPU @ 2.30GHz
RAM	4GB

6.2 Preparation

After building the database, designing the tables, inserting and compiling data, and constructing the basic levels of the Arabic ontology tree in the database, a web-based application was built to be the platform on which the experiment will be conducted, this system was built using ASP.NET Framework in (MVC) style, this application also includes the functions and classes responsible for processing and analyzing Arabic text and keep the results in another file. But there are several factors that may affect the results which should be taken into account and focused on in this experiment and these factors called parameters they are as follows:

- Text content: the content type of the text means what the components and the subject of the text which may affect the accuracy of the results, because text that contains many symbols and information in the form of images will be difficult to analyze and therefore the results will not be accurate as required, also analyze Arabic text in certain field such as physics, biology, Arabic, poetry, and Islamic education, may yield results varying in accuracy. This means that Arabic text that contains many images will be difficult to extract accurate learning objectives because it is difficult to analyze the content of images accurately, and also the text containing the lines and points have been mentioned continuously in the form of a story without any ordinal points or symbols or without spaces and blank lines, it will also be difficult to get accurate learning objectives in such cases.

- Text size: Also the size of the text may have a role in the accuracy of the results, so that analyzing a text of a smaller size will give a more realistic and accurate result and vice versa, the size scale here is large and normal text. For example, when the Arabic text is large, the answers to some information may be in stages, and not all the answers are in the same sentence or the same paragraph, but part of the answer comes in later paragraphs or even in the last text. Therefore, it will be very difficult to link these paragraphs together to get one single answer, which may affect the accuracy of the extracted learning objectives.

The experiment will be conducted by inserting an Arabic text as a text file and will apply the functions and script to extract meaningful learning objectives, which include extracting the enumerations, extracting the definitions, extracting the comparisons, extracting cause and effect, extracting the verses of the Holy Quran, extracting the entities classified in the ontology tree and finally extracting the global terms, then the results will be recorded and measured,, and when we repeate the experiment several times we will get the same result after that we will insert another text with different content and recording the results again and so on, until all parameters taken into account in the experiment. Thus, the experiment will be conducted many times and in each time the results will be recorded and measured according to the different factors and parameters to analyze the effect of each parameter or factor on the results, finally the results will be categorized by learning objectives in each attempt and evaluated according to each category. Therefore, many scenarios will be planned to conduct the experiment based on the diversity of parameters.

6.3 Arabic Text Files and Scenarios

The Arabic text that we will use in the experiment is an Arabic text of school courses of various subjects, including science, history and Islamic courses.

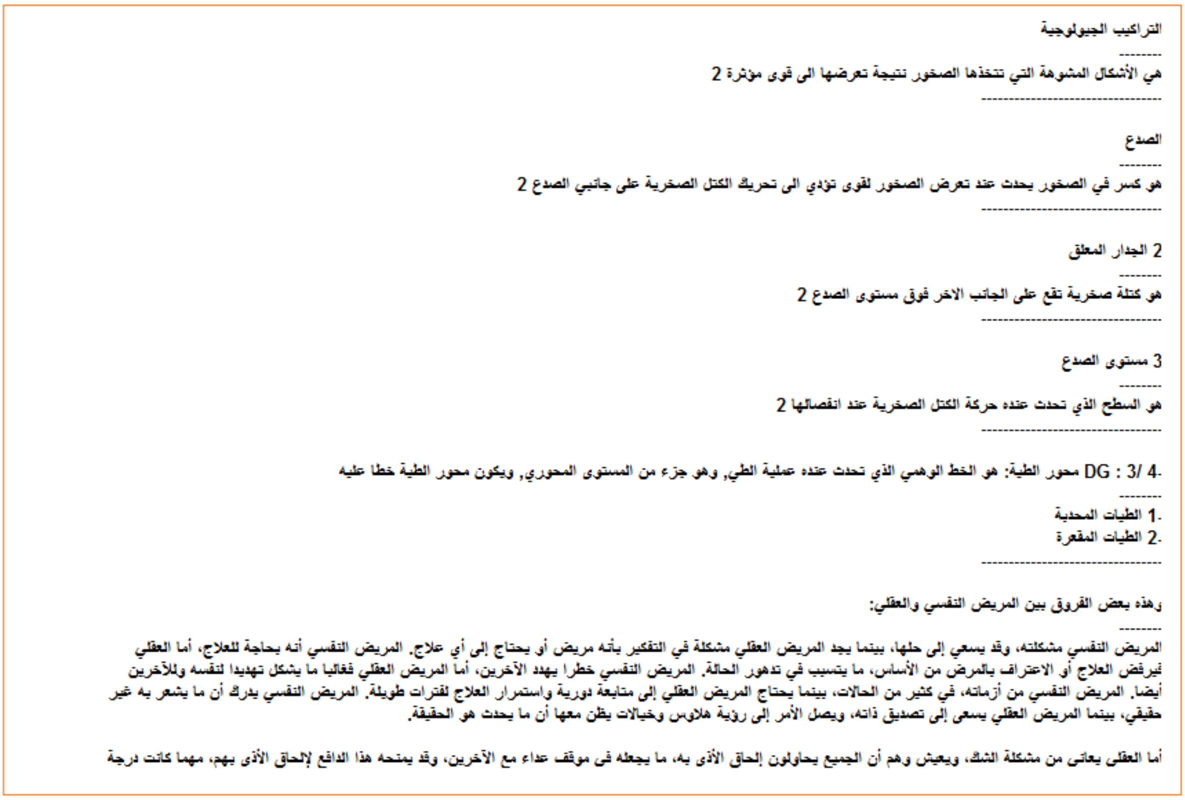
And the process of conducting the experiment will be within several scenarios so that each scenario will analyze a different content type of Arabic text with size approximately 5 pages, only 5 pages of each content can be satisfied for two reasons: First, because the extraction of a few results facilitates the evaluation process. Second, processing only five pages does not take long, but when processing a full book, the performance is less and it may take 10 minutes before obtaining results.

The content of the file is then read using the system and the system begins processing the content. The text is divided into paragraphs, sentences, and tokens and then normalizing the text before extraction process, after that he system then calls the the method that responsible for the extraction of each group of learning objective (definitions, enumerations etc ...).

After processing the text, the extracted results are stored in a file, which is learning objectives within the previously defined learning objectives groups. The experiment scenarios as follows:

6.3.1 Sience Content.

When entering text content from a science course about 5 pages in length, the results included definitions, enumurations and some comparison of terms and these results will be used to calculate Precision, Recall and F-measure as explained earlier, this is a section of results shown in Figure (6.1), but the full results will be appended to the appendix section.



التراكيب الجيولوجية

هي الأشكال المشوهة التي تتخذها الصخور نتيجة تعرضها الى قوى مؤثرة 2

الصدع

هو كسر في الصخور يحدث عند تعرض الصخور لقوى تؤدي الى تحريك الكتل الصخرية على جانبي الصدع 2

----

2 الجدار المعلق

هو كتلة صخرية تقع على الجانب الاخر فوق مستوى الصدع 2

----

3 مستوى الصدع

هو السطح الذي تحدث عنده حركة الكتل الصخرية عند انفصالها 2

----

4. 3/ : DG محور الطية: هو الخط الوهمي الذي تحدث عنده عملية الطي, وهو جزء من المستوى المحوري, ويكون محور الطية خطا عليه

----

1. الطيات المحدبة
2. الطيات المقعرة

وهذه بعض الفروق بين المريض النفسي والعقلي:

المريض النفسي مشكلته، وقد يسعي إلى حلها، بينما يجد المريض العقلي مشكلة في التفكير بأنه مريض أو يحتاج إلى أي علاج. المريض النفسي أنه بحاجة للعلاج، أما العقلي فيرفض العلاج أو الاعتراف بالمرض من الأساس، ما يتسبب في تدهور الحالة. المريض النفسي خطرا يهدد الآخرين، أما المريض العقلي فغالبا ما يشكل تهديدا لنفسه وللآخرين أيضا. المريض النفسي من أزماته، في كثير من الحالات، بينما يحتاج المريض العقلي إلى متابعة دورية واستمرار العلاج لفترات طويلة. المريض النفسي يدرك أن ما يشعر به غير حقيقي، بينما المريض العقلي يسعى إلى تصديق ذاته، ويصل الأمر إلى رؤية هلاوس وخيالات يظن معها أن ما يحدث هو الحقيقة.

أما العقلى يعانى من مشكلة الشك، ويعيش وهم أن الجميع يحاولون إلحاق الأذى به، ما يجعله فى موقف عداء مع الآخرين، وقد يمنحه هذا الدافع لإلحاق الأذى بهم، مهما كانت درجة

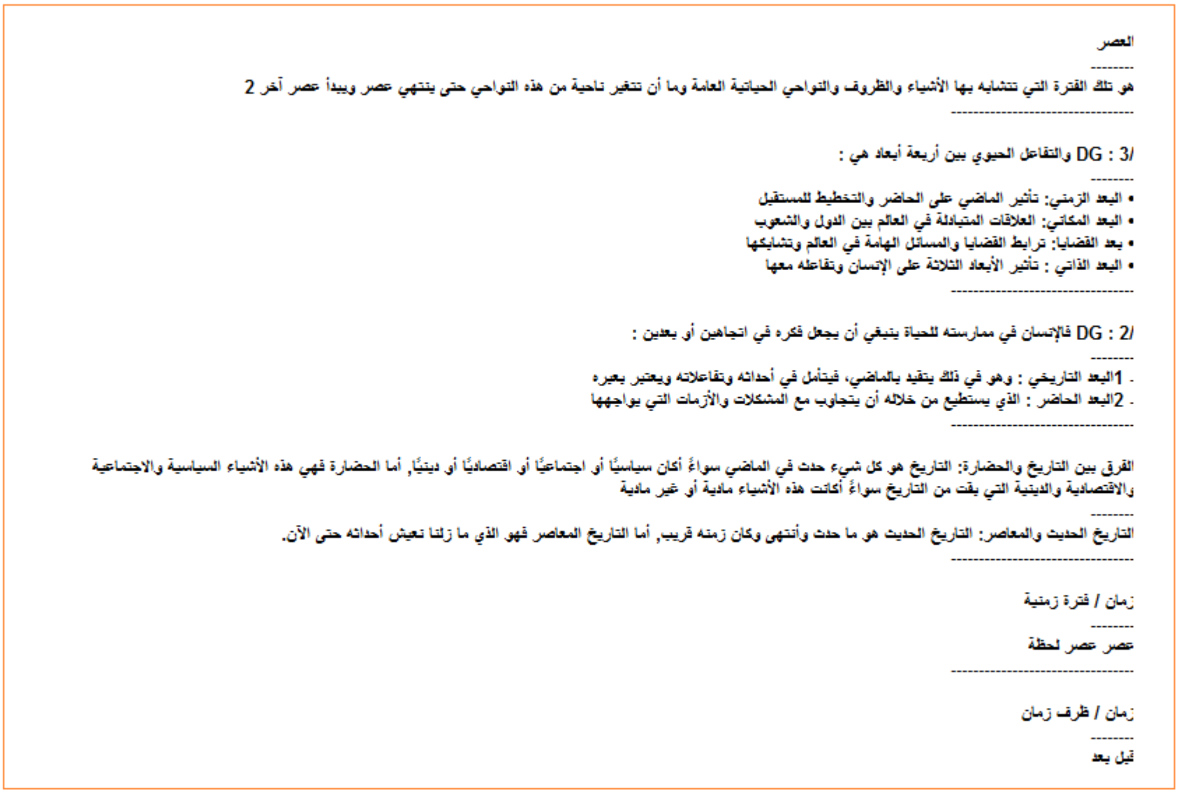
Figure (6.1): Results of Science Content

According to the results obtained in the above in details as follows:

- Definitions: the resulting number = 12, the actual number = 15.
- Enumerations: R = 2, A = 3.
- Comparisions: R = 1, A = 1.

6.3.2 Text content of history.

In this scenario, the Arabic text content of the course of history will be tested because of the difference in the terminology of the history of the terms of other fields to study the effect of the text content on the quality of the results.



```
العصر
--------
هو تلك الفترة التي تتشابه بها الأشياء والظروف والنواحي الحياتية العامة وما أن تتغير ناحية من هذه النواحي حتى ينتهي عصر ويبدأ عصر آخر 2
--------------------------------

DG : 3/ والتفاعل الحيوي بين أربعة أبعاد هي :
--------
• البعد الزمني: تأثير الماضي على الحاضر والتخطيط للمستقبل
• البعد المكاني: العلاقات المتبادلة في العالم بين الدول والشعوب
• بعد القضايا: ترابط القضايا والمسائل الهامة في العالم وتشابكها
• البعد الذاتي : تأثير الأبعاد الثلاثة على الإنسان وتفاعله معها
--------------------------------

DG : 2/ فالإنسان في ممارسته للحياة ينبغي أن يجعل فكره في اتجاهين أو بعدين :
--------
. 1البعد التاريخي : وهو في ذلك يتقيد بالماضي، فيتأمل في أحداثه وتفاعلاته ويعتبر بعبره
. 2البعد الحاضر : الذي يستطيع من خلاله أن يتجاوب مع المشكلات والأزمات التي يواجهها
--------------------------------

الفرق بين التاريخ والحضارة: التاريخ هو كل شيء حدث في الماضي سواءً أكان سياسيًا أو اجتماعيًا أو اقتصاديًا أو دينيًا, أما الحضارة فهي هذه الأشياء السياسية والاجتماعية
والاقتصادية والدينية التي بقت من التاريخ سواءً أكانت هذه الأشياء مادية أو غير مادية
--------
التاريخ الحديث والمعاصر: التاريخ الحديث هو ما حدث وأنتهى وكان زمنه قريب, أما التاريخ المعاصر فهو الذي ما زلنا نعيش أحداثه حتى الآن.
--------------------------------

زمان / فترة زمنية
--------
عصر عصر لحظة
--------------------------------

زمان / ظرف زمان
--------
قبل بعد
```

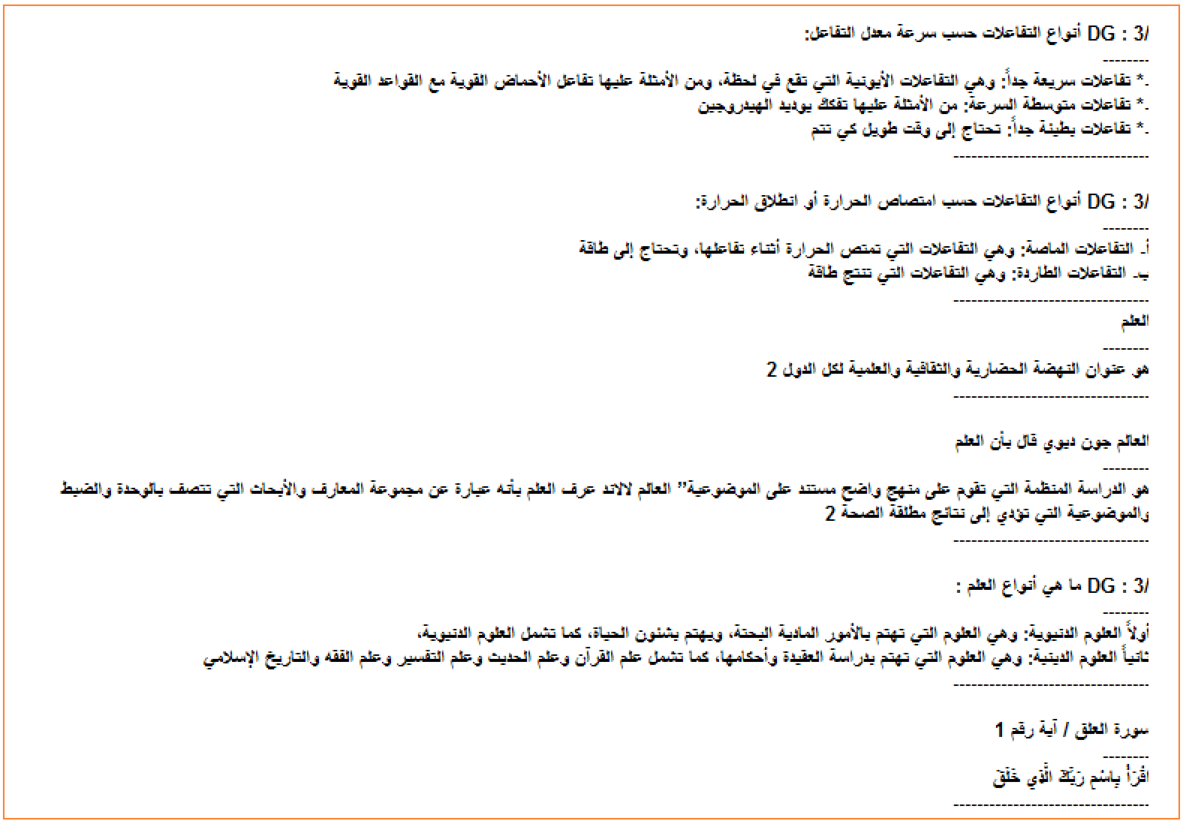
Figure (6.2): Results of History Content

The results obtained from history content as follows:

- Definitions: the resulting number = 8, the actual number = 7.
- Enumerations: R = 5, A = 4.
- Comparisions: R = 2, A = 2.
- Terms from ontology tree: R = 2, A = 2.

6.3.3 Text content of general topics.

In this scenario, the Arabic text contains various topics such as texts from the Koran and science, etc. And the main goal of the diversity of texts is to obtain a variety of learning objectives for comprehensive evaluation.



```
DG : 3/ أنواع التفاعلات حسب سرعة معدل التفاعل:
--------
.* تفاعلات سريعة جداً: وهي التفاعلات الأيونية التي تقع في لحظة، ومن الأمثلة عليها تفاعل الأحماض القوية مع القواعد القوية
.* تفاعلات متوسطة السرعة: من الأمثلة عليها تفكك يوديد الهيدروجين
.* تفاعلات بطيئة جداً: تحتاج إلى وقت طويل كي تتم
----------------------------------

DG : 3/ أنواع التفاعلات حسب امتصاص الحرارة أو انطلاق الحرارة:
--------
أـ التفاعلات الماصة: وهي التفاعلات التي تمتص الحرارة أثناء تفاعلها، وتحتاج إلى طاقة
بـ التفاعلات الطاردة: وهي التفاعلات التي تنتج طاقة
----------------------------------
العلم
--------
هو عنوان النهضة الحضارية والثقافية والعلمية لكل الدول 2
----------------------------------

العالم جون ديوي قال بأن العلم
--------
هو الدراسة المنظمة التي تقوم على منهج واضح مستند على الموضوعية" العالم لالاند عرف العلم بأنه عبارة عن مجموعة المعارف والأبحاث التي تتصف بالوحدة والضبط
والموضوعية التي تؤدي إلى نتائج مطلقة الصحة 2
----------------------------------

DG : 3/ ما هي أنواع العلم :
--------
أولاً العلوم الدنيوية: وهي العلوم التي تهتم بالأمور المادية البحتة، ويهتم بشئون الحياة، كما تشمل العلوم الدنيوية،
ثانياً العلوم الدينية: وهي العلوم التي تهتم بدراسة العقيدة وأحكامها، كما تشمل علم القرآن وعلم الحديث وعلم التفسير وعلم الفقه والتاريخ الإسلامي
----------------------------------

سورة العلق / آية رقم 1
--------
اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ
----------------------------------
```

Figure (6.3): Results of General Content.

The results obtained from general content as follows:

- Definitions: the resulting number = 6, the actual number = 4.
- Enumerations: R = 6, A = 7.
- Terms from ontology tree: R = 1, A = 1.
- Verses from the holy Quran: R = 1, A = 1.

For each of the above experiment scenarios, the experiment was repeated more than once and in each time we obtained the same result.

6.4 Evaluation of Results

As we mentioned above the experiment was carried out on texts in the Arabic language from a variety of subjects, and the extraction of all learning objectives in the text with their answers, which were previously defined either by the general template or based on Arabic Ontology or based on the data store of general terms and others.

After conducting the experiment, the results we obtained were evaluated by recall, precision and f-measure, for each set of learning objectives, we counted them in the original text and in the results file. We also calculated the number of correct results and the number of erroneous results for each set of learning objectives separately in the results file.

As mentioned in the previous chapters, precision, recall and F-measure are calculated using the following equations:

$$\text{Precision} = \frac{\text{True Extracted}}{\text{True Extracted} + \text{False Extracted}}$$

$$\text{Recall} = \frac{\text{True Extracted}}{\text{True Extracted} + \text{False Not Extracted}}$$

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

According the results obtained Science content:

A. **Case 1:** According the results obtained from science content:

- Definitions: the resulting number = 12, the actual number = 15.

  TP = 11, FP = 1, FN = 15 – 11 = 4.

  So

  Precision = 11 /11 + 1 = 91.

  Recall = 11 / 11 + 4 = 73.

  F-measure = 2 × (91 × 73 / 91 + 73) = 81

- Enumerations: Result = 2, Actual = 3.

  TP = 2, FP = 0, FN = 1

  Precision = 2 /2 = 100.

  Recall = 2 / 2 + 1 = 66.

  F-measure = 80.

- Comparisions: R = 1, A = 1.

  TP = 1, FP = 0, FN = 0

  Precision = 100.

  Recall = 100.

  F-measure = 100.

The average: Precision = 97, Recall = 80 and F-measure = 87

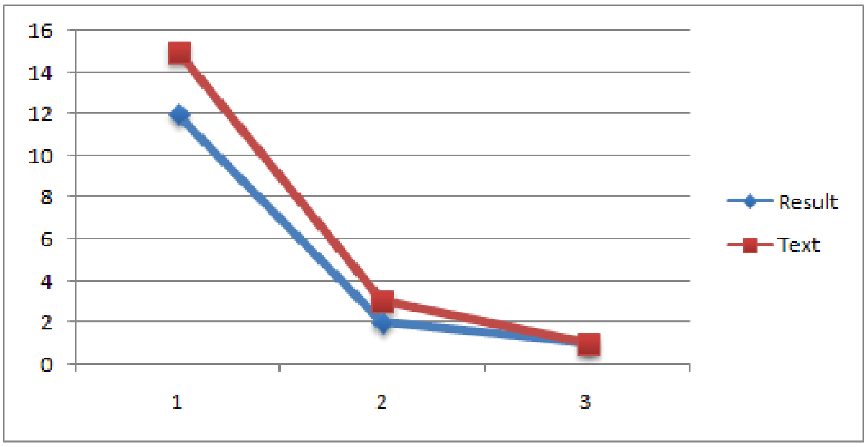


Figure (6.4): Curve of Science Text Results

B. **Case 2:** According the results obtained from history content:

- Definitions: the resulting number = 8, the actual number = 7.

  TP = 7, FP = 1, FN = 0

  Precision = 88.

  Recall = 100.

  F-measure = 94

- Enumerations: R = 5, A = 4.

  TP = 4, FP = 1, FN = 0

Precision = 80.

Recall = 100.

F-measure = 89.

- Comparisions: R = 2, A = 2.

TP = 2

Precision = 100.

Recall = 100.

F-measure = 100.

- Terms from ontology tree: R = 2, A = 2

Precision = 100.

Recall = 100.

F-measure = 100.

The average: Precision = 92, Recall = 100 and F-measure = 96.

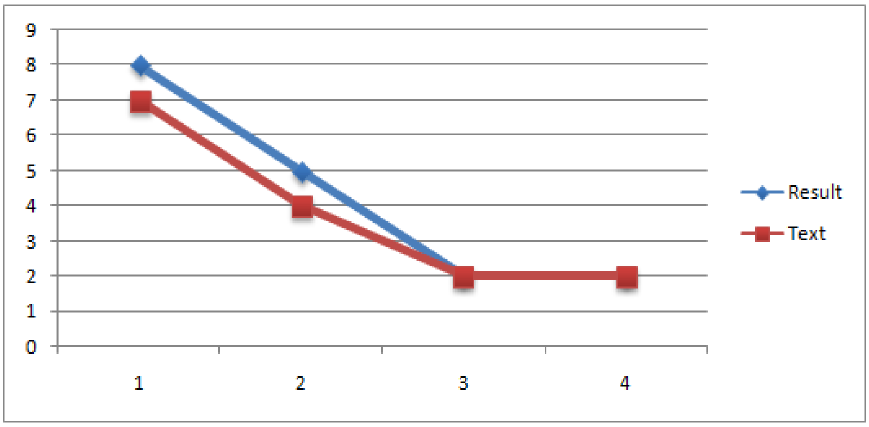


Figure (6.5): Curve of History Text Results

C. **Case 3:** According the results obtained from general content:

- Definitions: the resulting number = 6, the actual number = 4.

  TP = 4, FP = 2, FN = 0

  Precision = 67.

  Recall = 100.

  F-measure = 80.

- Enumerations: R = 6, A = 7.

  TP = 6, FP = 0, FN = 1

  Precision = 100.

  Recall = 86.

  F-measure = 92.

- Terms from ontology tree: R = 1, A = 1.

  Precision = 100, Recall = 100 and F-measure = 100

- Verses from the holy Quran: R = 1, A = 1.

  Precision = 100, Recall = 100 and F-measure = 100

The average: Precision = 92, Recall = 97 and F-measure = 93.

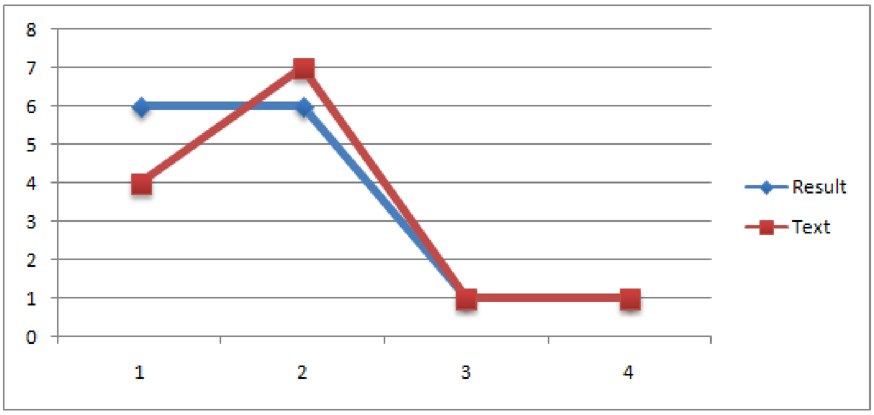


Figure (6.6): Curve of General Text Results

The results showed that the accuracy of extracting correct information is 94 for all text contents and the correct extraction completion rate is 92 and the balanced F-score = 92 as shown in Table (6.2).

Table (6.2): Summary of Precision, Rrecall and F-measure for All Text Contents

Learning objectives Groups	**Precision**	**Recall**	**F-measure**
All contents	94	92	92

6.5 Discussion

Through the results obtained we noticed that the precision = 94, recall = 92. And we obtained the highest percentages for definitions and comparisons and entities extracted from the ontology tree and the Quranic verses, in general, the lowest percentage of precision and recall was for learning objectives that were extracted based on general templates and the other groups of Learning objectives obtained the highest values of precision and recall, but in particular the enumerations have the lowest rates for several reasons:

- The extraction of the censuses should be based on a predefined general template that includes some indicators, symbols and conditions. This is what we mean by the degree of template matching (DTM). For example, within the conditions, the sentence should contain plural word and checking if the word is plural or single can be inaccurate.
- Enumerations may come within the text between the lines without any clear indications or numbering, in which case it will not be recognized by it's general template.

Through the results obtained, we noticed also that the quality of the results depends greatly on the quality of the text content as the scientific texts containing the symbols and images significantly will not be a suitable environment for extracting accurate learning objectives and exactly contrary to the literary texts in which the text coordinated and arranged

Consistently. And regarding the learning objectives that were more accurately extracted were those that were matched with predefined data such as the verses of the Holy Quran and global terms. We also note that if we increase the degree of template matching, the learning objectives extracted are more quality and less quantity.

Also, we should not ignore the role of other algorithms and their importance in obtaining accurate results. For example, accuracy in extracting enumurations depends mainly on the algorithm that works on examining the word if it is plural or not.

6.6 Summary

In this chapter we explained the equipment and preparations needed to conduct the experiment, including tools, programs and data. Then we explained how the experiment was conducted within specific scenarios, and then we recorded a sample of the obtained results to evaluate these results and to visualize them in a chart and then briefly summarize the evaluation results and finally discuss the results and interpret the convergence and the differences between them.

In the next chapter, we will summarize the thesis briefly and then identify key points for future work.

Chapter 7

Conclusions

Chapter 7

Conclusions

7.1 Summary

The extraction of educational objectives from textbooks and educational curricula requires the teacher to read the entire book in details, Therefore, this research focused on the process of automating the extraction of learning objectives in various subjects according to the first two levels of Bloom's taxonomy (remember and understand), To achieve this goal accurately and to answer the question of research effective answer, four basic stages have been adopted, The first is to build a tree that contains the words of the Arabic language and branch out based on the origins of the words and their relation to each other, Second, collect as many of the well-known global terms as possible and classify them under a key word, which will eventually be linked to the ontology tree, Third, the stage of processing the text in which the text is processed and dealt with in multiple ways according to the requirements of the other stages including the removal of the diacritical marks, the placement of the diacritical marks, splitting the text into paragraphs, sentences, words and tokens, normalization by removing the excess characters from the original word and finally checking the word if They was a plural or a verb, The final stage is by classifying the learning objectives that fall within the first and second level of the Bloom's taxonomy into groups and then building a general template for each of these groups to match the text based on these templates.

After applying these four stages on the Arabic text within the framework of the proposed approach in this thesis the result was extracting all the words that fall within the same classification in the ontology tree, extracting the global terminology mentioned in the text, and extracting the definitions and enumeration, extracting cause and effect statements, extracting the verses of Holly Quranic and extracting the comparisons between terms.

And regarding evaluation of the proposed approach in this thesis, the evaluation was using precision, recall and F-Measure, The results of the evaluation were promising for all groups of learning objectives under various factors and parameters, Where the evaluation results for extracting the words classified in the ontology tree was is precision 86% and recall is 90% and for extracting global terms the precision is 86% and recall is 90% and the results for extracting definitions was is precision 86% and recall is 90%, the results for extracting enumerations was precision is 86% and recall is 90%, the results for extracting cause and effect was precision is 86% and recall is 90%, the results for extracting the verses of holly Qur'an was precision is 86% and recall is 90%, the results for extracting comparisons was precision is 86% and recall is 90%

The result generally indicates that the approach used to answer the research question is an effective and very important approach.

7.2 Future Work

In this thesis we have done a long and hard work in order to get to what we have reached and in order to achieve valuable results, but nevertheless the work has not yet ended, and it lacks a lot of additions and developments if we want to achieve better and comprehensive results in the future.

And among these developments and additions we suggest the following: -

- Completing and enriching the ontology tree in words and details to reach wider and finer levels.
- Link the ontology tree to different knowledge areas, by classifying the words in the tree of the ontology by their scientific field, for example some words are linked in specific fields such as the word of poetry linked to the literature.
- Extracting learning objectives from advanced courses like medical curriculum.

References

[1] Motoei, A., François, C. and Juan, G, "How to apply the Bloom taxonomy to software engineering," *Software Technology and Engineering Practice, 2003. Eleventh Annual International Workshop on*, vol., no., pp.117, 122, 19- 21

[2] Mustafa, Jarrar, "Building a Formal Arabic Ontology (Invited Paper.)," in Proceedings of the Experts Meeting on Arabic ontologies & Semantic Networks Alecso, Arab League.Tunis, April 2011, pp.26-28.

[3] Belhassen, G. and Ramzi, F, "an approach to assist learners to identify their learning objectives in Personal Learning Environment (PLE)," Conference Paper, · December 2015.

[4] Silvia, EG. And Timothy, S, "Exploring Learning Objectives at Scale Through Concept Mapping of MOOC Learner Discussions," In Proceedings of the European MOOC Stakeholder Summit 2016.

[5] Sreeramana, A, "Student performance and Learning Outcomes in Higher Education Institutions," International Journal of Scientific Research & Modern Education (IJSRME), June 2016, pp.2455-5630.

[6] Mohammed, AL Zamil. And Qasem, A. Al-Radaideh, "Automatic Extraction of Ontological Relations from Arabic Text," Journal of King Saud University – Computer and Information Sciences (2014) 26, 462–472

[7] Zaid, N. Al-Shammari, "ANALYSIS MODEL FOR LEARNING OUTCOMES (AMLO)," Kuwait University, Conference Paper · May 2010.

[8] Tarek, Helmy. and Abdirahman, Daud, "Intelligent Agent for Information Extraction from Arabic Text without Machine Translation,"Tanta University, Egypt, Jan 2010.

[9] T. R. GRUBER, A translation approach to portable ontologies. Knowledge Acquisition, 5(2), pp. 199– 220, 1993.

[10] H. HERRE ET AL., General Formal Ontology (GFO): A Foundational Ontology Integrating Objects and Processes.Technical Report, 8, University of Leipzig, 2006.

[11] THOMAS C. JEPSEN, "This tutorial article describes some definitions of “ontology” as it relates to computer applications," published article by the IEEE Computer Society, o September/October 2009.

[12] JOHANNES BUSSE, BERNHARD HUMM and CHRISTOPH LUBBERT, "Actually, What Does “Ontology” Mean?" Journal of Computing and Information Technology- CIT, 23, 2015, 1, 29–41.

[13] G. PICKERT, Einfuhrung in Ontologien, ¨http://www.dbis.informatik.hu-berlin.de /dbisold/lehre/WS0203/SemWeb/artikel/2 /Pickert Ontologien final.pdf.

[14] Basel Salama El Zraie, "IExtraction of Taxonomic Relations from Arabic Text for Ontology Construction," The Islamic University–Gaza, Palestine, July 2016.

[15] https://en.wikipedia.org/wiki/Accuracy_and_precision, BS ISO 5725-1: "Accuracy (trueness and precision) of measurement methods and results - Part 1: General principles and definitions.", p.1 (1994).

[16] Microsoft, (10/9/2012). Microsoft .NET Framework 4.5 [.NET Framework 4.5 is a highly compatible, in-place update to .NET Framework 4.]. Retrieved from, https://www.microsoft.com/en-us/download/details.aspx?id=30653.

[17] Microsoft, (20/7/2015). Introduction to the C# Language and the .NET Framework. Retrieved from, https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework.

[18] Ibrahim Imam and Alaa Hamouda, “. An Ontology-based Summarization System for Arabic Documents (OSSAD)", International Journal of Computer Applications, Egypt, July 2013.

[19] Basel Salama El Zraie, "Extraction of Taxonomic Relations from Arabic Text for Ontology Construction," International Journal of Computer Applications, the Islamic University–Gaza, July 2016.

[20] Rinat Levy-Cohen, “Automated Extraction of Learning Goals and Objectives from Syllabi using LDA and Neural Nets,” International Journal of Computer Applications, Columbia University, March 2018.

[21] Arfaj, A., & Al Salman, A, ". Towards Ontology Construction from Arabic Texts-A Proposed Framework," Paper presented IEEE at International Conference on Computer and Information Technology (CIT). 2014.

[22] G. A. Miller, R. Beckwith, C. D. Fellbaum, D. Gross, K. Miller. 1990. WordNet: An online lexical database. Int. J. Lexicograph. 3, 4, pp. 235–244.

[23] Tengku Mohd T. Sembok, Belal Mustafa Abu Ata and Zainab Abu Bakar, "A Rule and Template Based Stemming Algorithm for Arabic Language," INTERNATIONAL JOURNAL OF MATHEMATICAL MODELS AND METHODS IN APPLIED SCIENCES. 2011.

[24] Hossam Ishkewy, Hany Harb and Hassan Farahat, "Azhary: An Arabic Lexical Ontology," Azhar University, International Journal of Web & Semantic Technology (IJWesT) Vol.5, No.4, October 2014.

[25] David M W Powers, "Evaluation: From Precision, Recall and F-Factorto ROC, Informedness, Markedness & Correlation," School of Informatics and Engineering Flinders University of South Australia, Technical Report SIE-07-001, December 2007.

[26] Cyril Goutte and Eric Gaussier, "A Probabilistic Interpretation of Precision, Recall and F-score, with Implication for Evaluation," Proceedings of the European Colloquium on IR Resarch (ECIR'05), LLNCS 3408 (Springer), pp.345–359.

[27] Marina Sokolova, Nathalie Japkowicz and Stan Szpakowicz, "Beyond Accuracy, F-score and ROC: A Family of Discriminant Measures for Performance Evaluation,".

[28] Majid Ziaratban, Karim Faez, Farhad Faradji, "Language-Based Feature Extraction Using Template-Matching in Farsi/Arabic Handwritten Numeral Recognition," Electrical

Engineering Department, Amirkabir University of Technology, Tehran, Iran, Conference Paper· September 2007, DOI: 10.1109/ICDAR.2007.4405576 IEEE.

[29] Iping Supriana*, Albadr Nasution, "Arabic Character Recognition System Development," Informatics Engineering, Institut Technology Bandung, Jalan Ganeca 10 Bandung 40132, Indonesia, The 4th International Conference on Electrical and Informatics (ICEEI 2013).

[30] Yang Xiang, Sarath Chandra Janga, "Building Integrated Ontological Knowledge Structures with Efficient Approximation Algorithms," Academic Editor: Dongchun Liang, BioMed Research International Volume 2015, Article ID 501528, 14 pages.

[31] Anwaya Aras, Sakshi Paratap, Dr. Mangesh Bedekar, "Ontological tree Generation for Enhanced Information Retrieval," Department of Computer Science, BITS-Pilani University, Goa, India, International Journal of Artificial Intelligence & Applications (IJAIA), Vol. 4, No. 4, July 2013 DOI: 10.5121/ijaia.2013.4405.

[32] GOBINDAG. CHOWDHUR, "Template Mining for Information Extraction from Digtal Documents," Gobinda G. Chowdhury, Division of Information Studies, School of Applied Science, Nanyang Technological [Jniversity, N4#2a-32, Nanyang Avenue, Singapore 639798 LIBRARY TRENDS, Vol. 48, No. I, Summer 1999, pp. 182-208.

[33] Jim Buckley and Chris Exton, "Blooms' Taxonomy: A Framework for Assessing Programmers' Knowledge of Software Systems," University of Limerick jim.buckley/chris.exton@ul.ie, DOI: 10.1109/WPC.2003.1199200 · Source: DBLP Conference: 11th International Workshop on Program Comprehension (IWPC 2003), May 10-11, 2003, Portland, Oregon, USA.

Appendix: code to extract learning objectives

A. Collect data

```
//insert verbs into database
 public void InsertVerbsIntoDB(string w)
        {
            LOB_DBEntities db = new LOB_DBEntities();
            string[] v = new string[3];
            v = ConvertPastToPresentAndFuture(w);
            if (v != null)
            {
                //dW = code.RemoveDiacritics(v[0]);
                string word = v[0];
                if (!db.LOB_Verbs.Any(x => x.verb_past.Equals(word)))
                {
                    var verbs = db.LOB_Verbs;
                    var obj = verbs.CreateObject();
                    obj.verb_past = v[0];
                    obj.verb_present_he = v[1];
                    obj.verb_future = v[2];
                    db.LOB_Verbs.AddObject(obj);
                    db.SaveChanges();
                }//
            }
        }
```

```
//insert nouns into database
 Public void InsertNounsIntoDB(string w = "", string plural = null, bool isMale = true,
long nodeId = 0 )
        {
            LOB_DBEntities db = new LOB_DBEntities();
            if (w != null)
            {
                if (!db.LOB_Nouns.Any(x => x.noun.Equals(w)))
                {
                    var verbs = db.LOB_Nouns;
                    var obj = verbs.CreateObject();
                    obj.noun = w;
                    obj.plural = plural;
                    obj.is_mle = isMale;
                    obj.node_id = nodeId;
                    db.LOB_Nouns.AddObject(obj);
                    db.SaveChanges();
                }//
            }
        }
//insert global terms into database
public void InsertTermsIntoDB(string w, string keyWord ="", string field = "")
        {
            LOB_DBEntities db = new LOB_DBEntities();
            if (w != null)
            {
                if (!db.LOB_General_Terms.Any(x => x.term.Equals(w)))
                {
                    var verbs = db.LOB_General_Terms;
                    var obj = verbs.CreateObject();
                    obj.term = w;
                    obj.keyword = keyWord;
                obj.term_field = field;

                db.LOB_General_Terms.AddObject(obj);

                db.SaveChanges();

            }//

        }

    }
```

B. Pre-processing Stage

```
//splitting text to paragraph
public List<string> SplitTextIntoParagraphs(string text)
{
List<string> result = new List<string>();
result = Regex.Split(text, "(\r\n){2,}").Where(x =>
!String.IsNullOrWhiteSpace(x)).ToList();
return result;
}
```

```
//splitting text to sentences
 public List<string> SplitTextIntoSentences(string text)
        {
            List<string> result = new List<string>();
            string[] sentencesDelimeters = new string[] { ";", ".", "؟","؛", "?", "!", "،", ","};
            result = text.Split(sentencesDelimeters,
StringSplitOptions.RemoveEmptyEntries).Where(x =>
!String.IsNullOrWhiteSpace(x)).ToList();
            return result;
        }
```

```
//splitting text to words
public List<string> SplitTextIntoWords(string text)
        {
            List<string> result = new List<string>();
            string[] wordsDelimiters = new string[]
            { " ", "|", "\\", "/", ".", ";", ",", "؛", "?", "!", "(", ")", "،", "[", "]", "{", "}", "\"",
"\'", "\t", "\n", "\r\n", "\n\n" };

            using (LOB_DBEntities db = new LOB_DBEntities())
            {
                var symboles = db.LOB_Symbols.Select(x => x.symbol).ToList();
                wordsDelimiters = wordsDelimiters.Concat(symboles).ToArray();
            }
            result = text.Split(wordsDelimiters, StringSplitOptions.RemoveEmptyEntries).
                Where(x => !String.IsNullOrWhiteSpace(x) &&
!String.IsNullOrEmpty(x)).ToList();
            return result;
        }
```

```
//splitting text to tokens
 public List<string> SplitTextIntoTokens(string text)
    {
      List<string> result = new List<string>();
      string[] sentencesDelimeters = new string[] { " ", "\t", "\n", "\r\n", "\n\n" };
      result = text.Split(sentencesDelimeters,
StringSplitOptions.RemoveEmptyEntries).Where(x =>
!String.IsNullOrWhiteSpace(x) && !String.IsNullOrEmpty(x)).ToList();
      return result;
    }



//removing diacritics Marks from text
 public string RemoveDiacritics(string text)
    {
      //String normalizedString = text.Normalize(NormalizationForm.FormKD);
      if (!String.IsNullOrEmpty(text))
      {
        String normalizedString = text;
        StringBuilder stringBuilder = new StringBuilder();
        for (int i = 0; i < normalizedString.Length; i++)
        {
          Char c = normalizedString[i];
          if (System.Globalization.CharUnicodeInfo.GetUnicodeCategory(c) !=
System.Globalization.UnicodeCategory.NonSpacingMark)
            stringBuilder.Append(c);
        }
        return stringBuilder.ToString();
      }//if
      else
      {
        return "";
      }//else
    }


// assign diacritics Marks on text (consuming API)
  public string Diacritics(string text)
   {
      //String normalizedString = text.Normalize(NormalizationForm.FormKD);
      if (!String.IsNullOrEmpty(text))
      {
        String result = string.Empty;
        String url
=@"https://api.multillect.com/translate/json/1.0/686?method=translate/api/vocalize&t
ext=" + text + "&sig=3da5b86f533d7f8491fba6bd19b50a5d";
        var client = new WebClient();
        var content = client.DownloadString(url);
        string[] array = content.Split(new char[] { "" });
        array[5] = array[5].Trim();
        result = Regex.Replace(array[5], @"\\u(?<Value>[a-fA-F0-9]{4})", m => {
return ((char)int.Parse(m.Groups["Value"].Value,
          NumberStyles.HexNumber)).ToString(); });
        return result;
      }//if
      else
      {
```

```
                return "";
            }//else
   }


//Read from file
string filePath = @"C:\fullpath\file.txt";
 string contents = File.ReadAllText(filePath);

//Write in file
string filePath = @"C:\fullpath\emptyfile.txt";
using (StreamWriter sw = File.AppendText(filePath))
{
    if (!String.IsNullOrEmpty(text))
    {
        sw.WriteLine(text.Trim());
    }
}




//Get the original word from the term
  public string GetOriginalWord(string text)
        {
            LOB_DBEntities db = new LOB_DBEntities();
            string originalWord = "", derivedWord = "", result = "" ;
            StringBuilder allHeadExtra = new StringBuilder();
            StringBuilder allTailExtra = new StringBuilder();

            Queue<string> headExtra = new Queue<string>();
            Queue<string> tailExtra = new Queue<string>();

            //string s = File.ReadAllText(@"fullpath\Nouns.txt");
            string[] nounsArray = db.LOB_Nouns.Select(x => x.noun).ToArray();

            string word = RemoveDiacritics(text);
            for (int H = 0; H < word.Length; H++)
            {
                for (int T = 0; T < word.Length - (1 + H); T++)
                {
                    derivedWord = word.Substring(H, word.Length - T - H);
                    if (Array.IndexOf(nounsArray, derivedWord) > -1)
                    {
                        originalWord = derivedWord;

                        for (int i = word.Length - T; i < word.Length; i++)
                            tailExtra.Enqueue(word[i].ToString());

                        for (int j = 0; j < H; j++)
                            headExtra.Enqueue(word[j].ToString());
                        break;
                    }//if
                }//for
            }//for

            //tailExtra.Reverse();
            while (tailExtra.Count > 0)
            {
                allTailExtra.Append(tailExtra.Dequeue());
            }

            while (headExtra.Count > 0)
                allHeadExtra.Append(headExtra.Dequeue());

            result = string.Format("The Original word is ({0}),\n the extra chars from the
head are {1}, \n and the extra chars from the tail are {2}", originalWord,
allHeadExtra, allTailExtra);
            //MessageBox.Show(string.Format("The Original word is ({0}),\n the extra
chars from the head are {1}, \n and the extra chars from the tail are {2}",
originalWord, allHeadExtra, allTailExtra));

            return result;
        }



//check the word is a verb or not
  public bool GetPastVerbs(string x)
  {
    //2 length   // more than 3 ......
     //ending with diphthong لَّ
            //string[] patternsOfPast = new string[] { "فَعَلْ", "أَفْعَلْ", "فَعَّلْ", "فاعَلْ", "إنْفَعَلْ",
"تَفَعَّلْ", "تَفَاْعَلْ", "إسْتَفْعَلْ", "فَعْلَلْ", "تَفَعْلَلْ", "إفْعَلَّ", "إفْعَاْلَّ", "إفْعَلَلَّ" };
            //string[] patternsOfPresent = new string[] { "يَفْعَلْ", "يُفْعِلْ", "يُفَعِّلْ", "يُفاعِلْ",
"يَنْفَعِلْ", "يَتَفَعَّلْ", "يَتَفَاْعَلْ", "يَسْتَفْعِلْ", "يُفَعْلِلْ", "يَتَفَعْلَلْ", "يَفْعَلُّ", "يَفْعَاْلُّ", "يَفْعَلِلُّ" };
            //string[] patternsOfFuture = new string[] { "إفْعَلْ", "أَفْعِلْ", "فَعِّلْ", "فاعِلْ", "إنْفَعِلْ",
"تَفَعَّلْ", "تَفَاْعَلْ", "إسْتَفْعِلْ", "فَعْلِلْ", "تَفَعْلَلْ", "إفْعَلَّ", "إفْعَاْلَّ", "إفْعَلِلَّ" };

            LOB_DBEntities db = new LOB_DBEntities();

            string fatha = db.LOB_Diacritics.FirstOrDefault(y => y.id == 4).diacritic;
            string damma = db.LOB_Diacritics.FirstOrDefault(y => y.id == 5).diacritic;
            string kasra = db.LOB_Diacritics.FirstOrDefault(y => y.id == 6).diacritic;
            string shadda = db.LOB_Diacritics.FirstOrDefault(y => y.id == 7).diacritic;
            string sukon = db.LOB_Diacritics.FirstOrDefault(y => y.id == 8).diacritic;

            char[] firstChar = new char[] { 'ا' ,'أ' ,'إ' };
            char[] present = new char[] {'ت','ي'};
            string pastWord = string.Empty;
            string dW = string.Empty;
            bool condition1 = false;
            bool condition2 = false;
            bool condition3 = false;
            bool condition4 = false;
            bool condition5 = false;
            bool condition6 = false;
            bool condition7 = false;
            bool condition8 = false;
            bool condition9 = false;
            bool condition10 = false;
            bool condition11 = false;
            bool condition12 = false;
            bool condition13 = false;
            bool condition14 = false;
            bool condition15 = false;
            bool condition16 = false;
            bool condition17 = false;

            bool result = false;



            if (!String.IsNullOrEmpty(x))
            {
                dW = RemoveDiacritics(x);
                //فَعَلَ وفَعِلَ وفَعُلَ  آزَرَ
                condition1 = (dW.Length == 3 && (x[1].ToString().Equals(fatha) ||
x[0].Equals('آ')) && (x[2].Equals('ا') || x[x.IndexOf(dW[1]) +
1].ToString().Equals(kasra) || x[x.IndexOf(dW[1]) + 1].ToString().Equals(fatha) ||
x[x.IndexOf(dW[1]) + 1].ToString().Equals(damma)) && !x.Contains(shadda) &&
(x[x.Length - 1].ToString().Equals(fatha) || x[x.Length - 1].Equals('ى') || x[x.Length -
1].Equals('ا')));

                // أَطْعَم أَفْعَل
                if (condition1 == false)
                    condition2 = (dW.Length == 4 && dW[0] == 'أ' &&
x[1].ToString().Equals(fatha) && ((x[3].ToString().Equals(sukon) &&
x[5].ToString().Equals(fatha)) || (x[3].ToString().Equals(fatha) && x[4].Equals('ا')))
&& (x[x.Length - 1].ToString().Equals(fatha) || x[x.Length - 1].Equals('ى') ||
x[x.Length - 1].Equals('ا')));

                //فَعَّلْ
                if (condition2 == false)
                    condition3 = (dW.Length == 3 && x[1].ToString().Equals(fatha) &&
x[3].ToString().Equals(shadda) && x[4].ToString().Equals(fatha) && (x[x.Length -
1].ToString().Equals(fatha) || x[x.Length - 1].Equals('ى') ) );

                //فاعَلْ  قَاتَلَ
                if (condition3 == false)
                    condition4 = (dW.Length == 4 && dW[1] == 'ا' &&
x[1].ToString().Equals(fatha) && x[4].ToString().Equals(fatha) && x[x.Length -
1].ToString().Equals(fatha));

                //إنْفَعَلْ && افْتَعَلَ
                if (condition4 == false)
                    condition5 = (dW.Length == 5 && dW[0] == 'إ' && x[x.IndexOf(dW[1])
+ 1].ToString().Equals(sukon) && x[x.IndexOf(dW[2]) + 1].ToString().Equals(fatha)
&& x[x.LastIndexOf(dW[3]) + 1].ToString().Equals(fatha) && (x[x.Length -
1].ToString().Equals(fatha) || x[x.Length - 1].Equals('ى') || x[x.Length -
1].Equals('ا')));

                //تَفَعَّلْ
                if (condition5 == false)
                    condition6 = (dW.Length == 4 && dW[0] == 'ت' &&
x[1].ToString().Equals(fatha) && x[3].ToString().Equals(fatha) &&
x[5].ToString().Equals(shadda) && x[6].ToString().Equals(fatha) && (x[x.Length -
1].ToString().Equals(fatha) || x[x.Length - 1].Equals('ى') ) );

                //تَفَاْعَلْ
                if (condition6 == false)
                condition7 = (dW.Length == 5 && dW[0] == 'ت' && dW[2] == 'ا' &&
x[1].ToString().Equals(fatha) && x[3].ToString().Equals(fatha) &&
x[6].ToString().Equals(fatha) && (x[x.Length - 1].ToString().Equals(fatha) ||
x[x.Length - 1].Equals('ى') ));

            //إسْتَفْعَلْ   إسْتَبْسَلَ   إسْتَعَان//
            if (condition7 == false)
                condition8 = (dW.Length == 6 && dW[0] == 'إ' && dW[1] == 'س' &&
x[x.IndexOf(dW[1]) + 1].ToString().Equals(sukon) && (x[x.IndexOf(dW[2]) +
1].ToString().Equals(fatha) || x[x.IndexOf(dW[2])].ToString().Equals("ا")) &&
x[x.IndexOf(dW[3]) + 1].ToString().Equals(sukon) && (x[x.LastIndexOf(dW[4]) +
1].ToString().Equals(fatha) || x[x.LastIndexOf(dW[4])].ToString().Equals("ا")) &&
x[x.Length - 1].ToString().Equals(fatha));

            //فَعْلَلْ//
            if (condition8 == false)
                condition9 = (dW.Length == 4 && dW[0] != 'أ' &&
!present.Contains(dW[0]) && dW[0] != 'م' && x[1].ToString().Equals(fatha) &&
x[3].ToString().Equals(sukon) && x[5].ToString().Equals(fatha) && x[x.Length -
1].ToString().Equals(fatha) && !x.Contains(shadda) && dW[dW.Length - 1 ] != 'ة');

            //تَفَعْلَلْ//
            if (condition9 == false)
                condition10 = (dW.Length == 5 && dW[0] == 'ت' &&
x[1].ToString().Equals(fatha) && x[3].ToString().Equals(fatha) &&
x[5].ToString().Equals(sukon) && x[7].ToString().Equals(fatha) && x[x.Length -
1].ToString().Equals(fatha));

            //إفْعَلَّ//
            if (condition10 == false)
                condition11 = (dW.Length == 4 && dW[0] == 'إ' &&
x[x.IndexOf(dW[1]) + 1].ToString().Equals(sukon) && x[x.IndexOf(dW[2]) +
1].ToString().Equals(fatha) && x[x.Length - 1].ToString().Equals(fatha) &&
x[x.Length - 2].ToString().Equals(shadda));

            //إفْعَالَّ//
            //condition12 = (dW.Length == 5 && dW[0] == 'إ' && dW[dW.Length - 2]
== 'ا' && x[1].ToString().Equals(fatha) && x[x.IndexOf(dW[1]) +
1].ToString().Equals(sukon) && x[x.IndexOf(dW[2]) + 1].ToString().Equals(kasra));

            //إفْعَلَلَّ//
            if (condition11 == false)
                condition13 = (dW.Length == 5 && dW[0] == 'إ' &&
x[x.IndexOf(dW[1]) + 1].ToString().Equals(sukon) && x[x.IndexOf(dW[2]) +
1].ToString().Equals(fatha) && x[x.IndexOf(dW[3]) + 1].ToString().Equals(fatha)
&& x[x.Length - 1].ToString().Equals(fatha) && x[x.Length -
2].ToString().Equals(shadda));

            // فَلَّ //
            if (condition13 == false)
                condition14 = (dW.Length == 2 && x[0] != 'أ' &&
x[1].ToString().Equals(fatha) && x[x.Length - 2].ToString().Equals(shadda) &&
x[x.Length - 1].ToString().Equals(fatha));

            // أَعَدَّ //
            if (condition14 == false)
```

```
                condition15 = (dW.Length == 3 && dW[0] == 'أ' &&
x[1].ToString().Equals(fatha) && x[3].ToString().Equals(fatha) && x[x.Length -
2].ToString().Equals(shadda) && x[x.Length - 1].ToString().Equals(fatha));

             if (condition15 == false)
                condition16 = (dW.Length == 3 && x[1].ToString().Equals(fatha) &&
x[3].ToString().Equals(fatha) && !x.Contains(shadda) && x[x.Length -
1].Equals('ى'));

             //إنَّعَلْ && اتَّعَلَ   اِتَّهَم  اِنَّمس//
             if (condition16 == false)
                condition17 = (dW.Length == 4 && dW[0] == 'إ' &&
x[x.IndexOf(dW[1]) + 1].ToString().Equals(shadda) && x[x.IndexOf(dW[1]) +
2].ToString().Equals(fatha) && x[x.IndexOf(dW[2]) + 1].ToString().Equals(fatha)
&& (x[x.Length - 1].ToString().Equals(fatha) || x[x.Length - 1].Equals('ى') ||
x[x.Length - 1].Equals('ا')));
          }

          result = (condition1 == true || condition2 == true || condition3 == true ||
condition4 == true || condition5 == true || condition6 == true || condition7 == true ||
condition8 == true || condition9 == true || condition10 == true || condition11 == true ||
condition12 == true || condition13 == true || condition14 == true || condition15 == true
|| condition16 == true || condition17 == true );
          //if (result == true)
          //    pastWord = x;


          return result;
       }


// check the word is a plural or not

public bool isPlural(string word)
       {
          // when conver past to present if past begin with hamza (إ\أ) or waw changed to
yaa and if the next also yaaa converted to waw like أيقـن يوقن



          // جموع التكسير
          // هو إما جمع قلة ويكون لما لا يزيد على العشرة وإما جمع كثرة و هو لما فوق العشرة
          // جموع القلة أربعة أوزان جمعت في قول بعضهم
          string[] patternsOfPlural = new string[] { "أَفْعُلٍ", "أَفعالٍ", "أَفْعِلةٍ", "فِعْلَةٍ", "", "", "",
"", "", "" };

          //جمع الكثرة//                                        [ for subject ] like قضاة وسحرة
          string[] patternsOfPlural2 = new string[] { "فُعْل", "فُعُل", "فُعَل", "فِعَل", "فُعَلَة", "
"فَعَلَة""", "فَعْلى", "فِعَلَة", "فُعَّل", "فُعَّال", "فِعال", "فُعول", "فِعْلان", "فُعْلان", "فُعَلاءُ", "أَفْعِلاءُ", "", "" };

          //extremely plural  صيغه منتهى الجموع:
          string[] patternsOfExtremelyPlural = new string[] { "فعالِل", "فعاليل", "أَفاعِل", "
"أَفاعيل", "تفاعِل", "تفاعِيل", "مفاعِل", "مفاعيل", "فواعِل", "فواعيل", "فعائل", "فُعالى", "فَعالِي",
"فعالـيُّ", "فُعالى""", "", "", "" };

          /*
           *
           *
```

```
 */

char[] firstChar = new char[] {'ت' ,'ي'};

LOB_DBEntities db = new LOB_DBEntities();
string fatha = db.LOB_Diacritics.FirstOrDefault(y => y.id == 4).diacritic;
string damma = db.LOB_Diacritics.FirstOrDefault(y => y.id == 5).diacritic;
string kasra = db.LOB_Diacritics.FirstOrDefault(y => y.id == 6).diacritic;
string shadda = db.LOB_Diacritics.FirstOrDefault(y => y.id == 7).diacritic;
string sukon = db.LOB_Diacritics.FirstOrDefault(y => y.id == 8).diacritic;
string kasratan = db.LOB_Diacritics.FirstOrDefault(y => y.id == 3).diacritic;

var numbers = db.LOB_Numbers.ToList();


string dW = string.Empty;
int startIndex = 0;
string[] res = new string[2];
string single = string.Empty;
StringBuilder sing = new StringBuilder();

bool condition1 = false;
bool condition2 = false;
bool condition3 = false;
bool condition4 = false;
bool condition5 = false;
bool condition6 = false;
bool condition7 = false;
bool condition8 = false;
bool condition9 = false;
bool condition10 = false;
bool condition11 = false;
bool condition12 = false;
bool condition13 = false;
bool condition14 = false;
bool condition15 = false;
bool condition16 = false;
bool condition17 = false;
bool condition18 = false;
bool condition19 = false;
bool condition20 = false;
bool condition21 = false;
bool condition22 = false;
bool condition23 = false;
bool condition24 = false;
bool condition25 = false;
bool condition26 = false;
bool condition27 = false;
bool condition28 = false;
bool condition29 = false;
bool condition30 = false;
bool condition31 = false;
bool condition32 = false;
bool condition33 = false;
bool condition34 = false;
bool result = false;
```

```
        if (!String.IsNullOrEmpty(word))
        {
            dW = RemoveDiacritics(word);
            dW = dW.StartsWith("ال") ? dW.Substring(2, dW.Length - 2) : dW;
            //جمع  المذكر والمؤنث السالم
            condition1 = (dW.Length > 4 &&
                (dW[dW.Length - 1] == 'ن' && dW[dW.Length - 2] == 'ي' &&
word[word.LastIndexOf(dW[dW.Length - 2]) + 1].ToString() != sukon &&
!firstChar.Contains(dW[0]) && !numbers.Any(x =>
RemoveDiacritics(x.enum1).Contains(dW) ||
RemoveDiacritics(x.enum2).Contains(dW) ||
RemoveDiacritics(x.enum3).Contains(dW) ||
RemoveDiacritics(x.text).Contains(dW)))
                ||
                (dW[dW.Length - 1] == 'ت' && dW[dW.Length - 2] == 'ا'  &&
!firstChar.Contains(dW[0]) && !numbers.Any(x =>
RemoveDiacritics(x.enum1).Contains(dW) ||
RemoveDiacritics(x.enum2).Contains(dW) ||
RemoveDiacritics(x.enum3).Contains(dW) || RemoveDiacritics(x.text).Contains(dW))
)
                ||
                (dW[dW.Length - 1] == 'ن' && dW[dW.Length - 2] == 'و' )
                && !firstChar.Contains(dW[0]) && !numbers.Any(x =>
RemoveDiacritics(x.enum1).Contains(dW) ||
RemoveDiacritics(x.enum2).Contains(dW) ||
RemoveDiacritics(x.enum3).Contains(dW) || RemoveDiacritics(x.text).Contains(dW))
);



            //أَفْعُلٍ // أذرع
            if (condition1 == false)
                condition2 = (dW.Length == 4 && dW[0] == 'أ' &&
word[word.IndexOf(dW[1]) + 1].ToString().Equals(sukon) &&
word[word.IndexOf(dW[2]) + 1].ToString().Equals(damma) );
            else
            {
                startIndex = word.LastIndexOf(dW[dW.Length - 2]);
                startIndex = word.Length - startIndex;
                single = word.Substring(0, word.Length - startIndex);
                res[0] = single;
                res[1] = word;
            }

            //أَفعالِ  أقفال
            if (condition2 == false)
                condition3 = (dW.Length == 5 && dW[0] == 'أ' && dW[dW.Length - 2]
== 'ا' );
            //else
            //{
            //    sing.Append(dW[1]);
            //    sing.Append(kasra);
            //    sing.Append(dW[2]);
            //    sing.Append("ا");
            //    sing.Append(dW[dW.Length - 1]);
            //    single = sing.ToString();
            //    res[0] = single;
        //    res[1] = word;
        //}

        //أَفْعِلةٍ وأَطعمة  أَعِزَّة
        if (condition3 == false)
          condition4 = ((dW.Length == 5 && dW[0] == 'أ' && dW[dW.Length - 1]
== 'ة') || (dW.Length == 4 && dW[0] == 'أ'  && word[word.IndexOf(dW[2]) +
1].ToString().Equals(shadda) && dW[dW.Length - 1] == 'ة'));

        //فِعْلةٍ  فتية
        if (condition4 == false)
          condition5 = (dW.Length == 4 && word[word.IndexOf(dW[0]) +
1].ToString().Equals(kasra) && dW[dW.Length - 1] == 'ة');


        //فُعْل  خُضْر   //  فُعُل  غُبُر  //فُعَل  وغُرف
        if (condition5 == false)
          condition6 = (dW.Length == 3 && word[word.IndexOf(dW[0]) +
1].ToString().Equals(damma) && (word[word.IndexOf(dW[1]) +
1].ToString().Equals(sukon) || word[word.IndexOf(dW[1]) +
1].ToString().Equals(damma) || word[word.IndexOf(dW[1]) +
1].ToString().Equals(fatha)));


        //قِطَع  فِعَل
        if (condition6 == false)
          condition7 = (dW.Length == 3 && word[word.IndexOf(dW[0]) +
1].ToString().Equals(kasra) && word[word.IndexOf(dW[1]) +
1].ToString().Equals(fatha));


        //قضاة  غزاة  فُعَلة
        if (condition7 == false)
          condition8 = (dW.Length == 4 && word[word.IndexOf(dW[0]) +
1].ToString().Equals(damma) && dW[dW.Length - 2] == 'ا' && dW[dW.Length - 1]
== 'ة');

        //فَعَلة  سَحَرة
        if (condition8 == false)
          condition9 = (dW.Length == 4 && word[word.IndexOf(dW[0]) +
1].ToString().Equals(fatha) && word[word.IndexOf(dW[1]) +
1].ToString().Equals(fatha) && dW[dW.Length - 1] == 'ة');

        //فَعْلى  مرضى
        if (condition9 == false)
          condition10 = (dW.Length == 4 && word[word.IndexOf(dW[0]) +
1].ToString().Equals(fatha) && word[word.IndexOf(dW[1]) +
1].ToString().Equals(sukon) && dW[dW.Length - 1] == 'ى');

        //فِعَلة  دببة
        if (condition10 == false)
          condition11 = (dW.Length == 4 && word[word.IndexOf(dW[0]) +
1].ToString().Equals(kasra) && word[word.IndexOf(dW[1]) +
1].ToString().Equals(fatha) && dW[dW.Length - 1] == 'ة');

        //فُعَّل  رُكَّع  غزَّى
        if (condition11 == false)
```

```
            condition12 = (dW.Length == 3 && word[word.IndexOf(dW[0]) +
1].ToString().Equals(damma) && word[word.IndexOf(dW[1]) +
1].ToString().Equals(shadda) );

          //فُعَّال  كتَّاب
          if (condition12 == false)
            condition13 = (dW.Length == 4 && word[word.IndexOf(dW[0]) +
1].ToString().Equals(damma) && word[word.IndexOf(dW[1]) +
1].ToString().Equals(shadda) && word[word.IndexOf(dW[1]) +
2].ToString().Equals(fatha) && dW[dW.Length - 2] == 'ا');

          //فِعال صِعاب جبال
          if (condition13 == false)
            condition14 = (dW.Length == 4 && word[word.IndexOf(dW[0]) +
1].ToString().Equals(kasra) && dW[dW.Length - 2] == 'ا');

          //فُعول  قلوب
          if (condition14 == false)
            condition15 = (dW.Length == 4 && word[word.IndexOf(dW[0]) +
1].ToString().Equals(damma) && dW[dW.Length - 2] == 'و');

          //فِعْلان  غِربان
          if (condition15 == false)
            condition16 = (dW.Length == 5 && word[word.IndexOf(dW[0]) +
1].ToString().Equals(kasra) && dW[dW.Length - 2] == 'ا' && dW[dW.Length - 1]
== 'ن');

          //فُعْلان  قُضبان
          if (condition16 == false)
            condition17 = (dW.Length == 5 && word[word.IndexOf(dW[0]) +
1].ToString().Equals(damma) && dW[dW.Length - 2] == 'ا' && dW[dW.Length - 1]
== 'ن');

          //فُعَلاءُ  كُرماء  جُلَساء
          if (condition17 == false)
            condition18 = (dW.Length == 5 && word[word.IndexOf(dW[0]) +
1].ToString().Equals(damma) && dW[dW.Length - 2] == 'ا' && dW[dW.Length - 1]
== 'ء');


          //أَفْعِلاءُ  أنبياء  أَطبّاء.
          if (condition18 == false)
            condition19 = ((dW.Length == 6 || (dW.Length == 5 &&
word[word.IndexOf(dW[2]) + 1].ToString().Equals(shadda))) && dW[0] == 'أ' &&
dW[dW.Length - 2] == 'ا' && dW[dW.Length - 1] == 'ء');

          //extremely plural  صيغه منتهى الجموع:
         // string[] patternsOfExtremelyPlural = new string[] { "فعالِل", "فعاليل", "أَفاعِل", "
أَفاعيل", "تفاعِل", "تفاعِيل", "مفاعِل", "مفاعيل", "فواعِل", "فواعيل", "فعائل", "فَعالى", "فَعالِي",
"فعالِيُّ", "فُعالى""", """, """, """ };

          //فعالِل  دراهم.
          if (condition19 == false)
            condition20 = (dW.Length == 5 && dW[2] == 'ا' &&
word[word.IndexOf(dW[3]) + 1].ToString().Equals(kasra));

          //فعاليل  دنانير.
```

```
            if (condition20 == false)
               condition21 = (dW.Length == 6 && dW[2] == 'ا' && dW[dW.Length -
2] == 'ي');

            //أَفاعِل  أَفاضل.
            if (condition21 == false)
               condition22 = (dW.Length == 5 && dW[0] == 'أ' && dW[2] == 'ا' &&
word[word.IndexOf(dW[3]) + 1].ToString().Equals(kasra) );

            //أَفاعيل  أَساليب.
            if (condition22 == false)
               condition23 = (dW.Length == 6 && dW[0] == 'أ' && dW[2] == 'ا' &&
dW[dW.Length - 2] == 'ي');

            //تفاعِل  تجارب.
            if (condition23 == false)
               condition24 = (dW.Length == 5 && dW[0] == 'ت' && dW[2] == 'ا' &&
word[word.IndexOf(dW[3]) + 1].ToString().Equals(kasra));

            //تفاعِيل  تسابيح.
            if (condition24 == false)
               condition25 = (dW.Length == 6 && dW[0] == 'ت' && dW[2] == 'ا' &&
dW[dW.Length - 2] == 'ي');

            //مفاعِل  مساجد.
            if (condition25 == false)
               condition26 = (dW.Length == 5 && dW[0] == 'م' && dW[2] == 'ا' &&
word[word.IndexOf(dW[3]) + 1].ToString().Equals(kasra));

            //مفاعيل  مصابيح.
            if (condition26 == false)
               condition27 = (dW.Length == 6 && dW[0] == 'م' && dW[2] == 'ا' &&
dW[dW.Length - 2] == 'ي' );

            //فواعِل  خواتم.
            if (condition27 == false)
               condition28 = (dW.Length == 5 && dW[1] == 'و' && dW[2] == 'ا' &&
word[word.IndexOf(dW[3]) + 1].ToString().Equals(kasra));

            //فواعيل  طواحين.
            if (condition28 == false)
               condition29 = (dW.Length == 6 && dW[1] == 'و' && dW[2] == 'ا' &&
dW[dW.Length - 2] == 'ي');

            //فعائل  لطائف. كرائم
            if (condition29 == false)
               condition30 = (dW.Length == 5 && dW[2] == 'ا' && dW[3] == 'ء' &&
word[word.IndexOf(dW[3]) + 1].ToString().Equals(kasra));

            //فَعالى  عَذارى.
            if (condition30 == false)
               condition31 = ((dW.Length == 5 && word[1].ToString().Equals(fatha)
&& dW[2] == 'ا' && dW[dW.Length - 1] == 'ى') || (dW.Length == 4 &&
word[1].ToString().Equals(fatha) && dW[2] == 'ا' && word[word.Length -
1].ToString().Equals(kasratan) ) );

            //فَعالِي  تراقٍ  صحاري.
```

```
                if (condition31 == false)
                    condition32 = ((dW.Length == 4 && dW[2] == 'ا' && word[word.Length
- 1].ToString().Equals(kasratan)) || (dW.Length == 5 && dW[2] == 'ا' &&
dW[dW.Length - 1] == 'ي'));

                //فُعالى  سُكارى.  غُضابى
                if (condition32 == false)
                    condition33 = (dW.Length == 5 && word[1].ToString().Equals(damma)
&& dW[2] == 'ا' && dW[dW.Length - 1] == 'ى');

                //فعاليٌّ  كراسيّ.
                if (condition33 == false)
                    condition34 = (dW.Length == 5 && dW[2] == 'ا' && dW[dW.Length -
1] == 'ي' && word[word.IndexOf(dW[dW.Length - 1]) +
1].ToString().Equals(shadda) );

            }

            result = (condition1 == true || condition2 == true || condition3 == true ||
condition4 == true || condition5 == true || condition6 == true || condition7 == true ||
condition8 == true
                || condition9 == true || condition10 == true || condition11 == true ||
condition12 == true || condition13 == true || condition14 == true || condition15 == true
|| condition16 == true
                || condition17 == true || condition18 == true || condition19 == true ||
condition20 == true || condition21 == true || condition22 == true || condition23 == true
|| condition24 == true
                || condition25 == true || condition26 == true || condition27 == true ||
condition28 == true || condition29 == true || condition30 == true || condition31 == true
|| condition32 == true
                || condition33 == true || condition34 == true );
            //if (result == true)
            //    return x;

            return result;
        }
```

C. Extracting Learning objectives based on Arabic ontology tree

```
public Dictionary<string, string> ExtractFromTree()
        {
            LOB_DBEntities db = new LOB_DBEntities();

            string result = "";
            string[] sentencesDelimeters = new string[] { ";", ".", "؛", "?", "!", ":" };
            List<long> nodesId = new List<long>();
            StringBuilder term = new StringBuilder();
            StringBuilder answer = new StringBuilder();
            Dictionary<string, string> termAnswer = new Dictionary<string, string>();
            var words = SplitTextIntoWords(ReadFromFileALL()).ToList();
            words = words.Select(x => RemoveDiacritics(x.Trim())).ToList();
            var nouns = db.LOB_Nouns.Where(x => x.noun != null).ToList();
            var list = nouns.Where(x => words.Any(y => y == x.noun.Trim())).ToList();
            //var list = nouns.SelectMany(x => x.noun).Select(g => new { Name = g.,
Count = g.Count() }).ToList();

            foreach (var word in list)
            {
                term.Clear();
                answer.Clear();
                try
                {
                    if (list.Count(x => x.node_id == word.node_id) > 1 &&
!nodesId.Contains((long)word.node_id))
                    {
                        nodesId.Add((long)word.node_id);
                        var level = db.LOB_Words_Tree.FirstOrDefault(x => x.id ==
word.node_id);
                        if (level != null)
                        {
                            var superLevel = db.LOB_Words_Tree.FirstOrDefault(x => x.id ==
level.super_id);
                            if (superLevel != null)
                            {
                                term.Append(superLevel.noun_ar + " / " + level.noun_ar);
                            }//if
                        }//if
                        //result = list.Count() + "  k";

                        var data = list.Where(x => x.node_id == word.node_id && x.noun !=
null).Select(x => x.noun);
                        //var data2 = list.Where(x => x.node_id == word.node_id &&
x.compound_noun != null).Select(x => x.compound_noun);
                        //data = data.Concat(data2);
                        foreach (var item in data)
                        {
                            if (!String.IsNullOrEmpty(item))
                            {
                                answer.Append(item + " ");
                            }
                        }

                        if (term.Length > 0 && answer.Length > 0)
                        {
                            if (!termAnswer.ContainsKey(term.ToString()))
                            {
                                termAnswer.Add(term.ToString(), answer.ToString());
                            }
                        }
                    }//if
                    //result = indicators + "  DG: " + degree + "  indx: " + wordIndex +
"sentence : " + sent + " ";
                }//try
                catch (Exception ex) { result = ex.Message; }
            }//for each sent
            return termAnswer;
        }
```

D. Extracting global terminologies

```
public Dictionary<string, string> ExtractGeneralTerms()
    {
      LOB_DBEntities db = new LOB_DBEntities();
      string result = "";
      string dW = "";
      StringBuilder term = new StringBuilder();
      StringBuilder answer = new StringBuilder();
      Dictionary<string, string> termAnswer = new Dictionary<string, string>();

      var text = ReadFromFileALL();
      dW = RemoveDiacritics(text);
      var nouns = db.LOB_General_Terms.Where(x => x.keyword !=
null).ToList();
      var list = nouns.Where(x =>
dW.Contains(RemoveDiacritics(x.term.Trim()))).ToList();
      foreach (var str in list)
      {
        term.Clear();
        answer.Clear();
        try
        {
          if (!String.IsNullOrEmpty(str.term) &&
!termAnswer.ContainsKey(str.keyword.Trim()))
          {
            term.Append(str.keyword.Trim());
            var data = list.Where(x => x.keyword == str.keyword).Select(x =>
x.term);
            foreach (var item in data)
            {
              if (!String.IsNullOrEmpty(item))
              {
                answer.Append(item.Trim() + " / ");
              }
            }//foreach

            if (term.Length > 0 && answer.Length > 0)
            {
              if (!termAnswer.ContainsKey(term.ToString().Trim()))
              {
                termAnswer.Add(term.ToString().Trim(),
answer.ToString().Trim());
              }
            }//if
          }//if
        }//try
        catch (Exception ex) { result = ex.Message; }
      }//for each sent
      return termAnswer;
    }
```

E. Extracting Learning objectives based on general templates

```
// Extract definitions
 public Dictionary<string, string> ExtractDefinitions()
      {
         string result = "";
         string dW = "";
         int nouns = 0;
         int degree = 0;
         int index = 0;
         string indicators = "";
         string[] definitionsIndicators = new string[] { "هو", "هي", "يعني", "تعني", "بانه",
"بانها", "المقصود ب" };
         StringBuilder term = new StringBuilder();
         StringBuilder definition = new StringBuilder();
         Dictionary<string, string> definAnswer = new Dictionary<string, string>();
         var content = ReadFromFileALL();
         var sentences = SplitTextIntoSentences(content);

         foreach(var sent in sentences)
         {
            indicators = "";
            term.Clear();
            definition.Clear();
            degree = 0;
            nouns = 0;
            index = 0;
            dW = RemoveDiacritics(sent);
            indicators = definitionsIndicators.Where(x =>
dW.Contains(x)).FirstOrDefault();
            try
            {
               if (!String.IsNullOrEmpty(indicators))
               {
                  degree++;// increment matching degree
                  var words = SplitTextIntoWords(sent);
                  index = words.IndexOf(indicators);
                  if (words[index - 1] == ":")
                  {
                     degree++;
                  }//contains :

                  if (words[index + 1] == "مصطلح")
                  {
                     degree++;
                  }//contains :

                  //extract the term
                  for (int i = 0; i < index; i++)
                  {
                     dW = "";
                     dW = RemoveDiacritics(words[i].Trim());
                     if (dW.StartsWith("ال"))
                     {
                        nouns++;
                     }
                     term.Append(words[i] + " ");
```

```
                        }
                        if (nouns > 0) { degree++; }

                        //extract the definition
                        for (int j = index; j < words.Count(); j++)
                        {
                            definition.Append(words[j] + " ");
                        }
                        definition.Append(degree);
                        if (degree > 1)
                        {
                            if (!definAnswer.ContainsKey(term.ToString()))
                            {
                                definAnswer.Add(term.ToString(), definition.ToString());
                            }
                        }
                    }// contains هو
                }//try
                catch (Exception ex) { result = ex.Message; }
            }//for each sent
            return definAnswer;
```

// Extract Enumerations

```
  public Dictionary<string, string> ExtractEnumerations()
        {
            LOB_DBEntities db = new LOB_DBEntities();
            var enum1 = db.LOB_Numbers.Take(20).ToArray(); // enum1
            var enum2 = enum1.Select(x => new {id = x.id, number = x.number, e1 =
RemoveDiacritics(x.enum1), e2 = RemoveDiacritics(x.enum2) }).ToArray();

            List<int> idList = new List<int>(){36, 48, 52, 58};
            var symbols = db.LOB_Symbols.Join(idList, up => up.id, ID => ID, (up, id)
=> up)
                .Select(x => new { id = x.id, symbol = x.symbol })
                .ToList();
            var enum3 = db.LOB_Arabic_Letters_Abjad.Select(x => new { id = x.id,
letter = x.letter} ).ToArray(); // enum4
            string result = "";
            string dW = "";
            int degree = 0;
            int indicatorIndex = 0;
            int sentIndex = 0;
            string indicators = "";
            string firstToken = "";
            List<string> definitionsIndicators = new List<string>() { " ,"يلي" ,"الآتية" ,"التالية"
وهي"", ":"};
            StringBuilder term = new StringBuilder();
            StringBuilder definition = new StringBuilder();
            Dictionary<string, string> termAnswer = new Dictionary<string, string>();
            var content = ReadFromFileALL();
            string[] sentencesDelimeters = new string[] { ";", ".", "؟" ,"!" ,"?" ,"؛",
Environment.NewLine + Environment.NewLine};
```

```
        var sentences = content.Split(sentencesDelimeters,
StringSplitOptions.RemoveEmptyEntries).Where(x =>
!String.IsNullOrWhiteSpace(x)).ToList();
        foreach (var sent in sentences)
        {
          indicators = "";
          term.Clear();
          definition.Clear();
          degree = 0;
          sentIndex = 0;
          indicatorIndex = 0;
          var words = SplitTextIntoWords(sent);
          dW = RemoveDiacritics(sent);
          indicators = definitionsIndicators.Where(x =>
dW.Contains(x)).FirstOrDefault();
          try
          {
            if (!String.IsNullOrEmpty(indicators))
            {
              degree++;// increment matching degree
              //if the term contains nouns, increment matching degree
              if (words.Any(x => RemoveDiacritics(x).StartsWith("ال") ||
RemoveDiacritics(x).StartsWith("وال")))
              {
                degree++;
              }
              //if the term contains plurals, increment matching degree
              if (words.Any(x => isPlural(x)))
              {
                degree++;
              }
            }// if contains indicators
            if (degree > 1)
            {
              //Search for successive sentences
              sentIndex = sentences.IndexOf(sent);
              int oc = 0; //counter to ensure that the enumeration is consecutive
              int i = 0;
              bool condition = false;
              while (sentIndex + 1 < sentences.Count())
              {
                firstToken = "";
                condition = false;
                string nextSent = sentences[sentIndex + 1];
                sentIndex = sentences.IndexOf(nextSent);
                if (!String.IsNullOrEmpty(nextSent))
                {
                  i++;
                  var wordsInNextSent = SplitTextIntoWords(nextSent);
                  firstToken = wordsInNextSent[0];
                  //check if first token the sentence is a numerical order or a symbol
                  condition = enum2.Any(x =>
x.e1.Contains(firstToken.ToString()) && (x.id == oc + 1)) || enum2.Any(x =>
x.e2.Contains(firstToken) && (x.id == oc + 1)) || enum2.Any(x =>
x.number.ToString().Contains(firstToken) && (x.id == oc + 1)) || enum3.Any(x =>
firstToken.Contains(x.letter) && (x.id == oc + 1)) || symbols.Any(x =>
firstToken.Contains(x.symbol));
```

```
                                if (condition == true)
                                {
                                    oc++;
                                    definition.Append(nextSent);
                                    definition.Append(Environment.NewLine);
                                    //result += firstToken + "sent : " + nextSent + "///";
                                }
                            }//if not empty
                        }//while not reach last of list
                        if (definition.Length > 0)
                        {
                            term.Append("DG : " + degree + "/ " + sent);
                            if (!termAnswer.ContainsKey(term.ToString()))
                            {
                                termAnswer.Add(term.ToString(), definition.ToString());
                            }
                        }//if definition not empty
                    }// if min matching degree achieved
                }//try
                catch (Exception ex) { result = ex.Message; }
            }//for each sent
            return termAnswer;


// Extract Comparisons
 public Dictionary<string, string> ExtractComparisons()
        {
            LOB_DBEntities db = new LOB_DBEntities();
            string result = "";
            string dW = "";
            int degree = 0;
            int index = 0;
            int sentIndex = 0;
            int wordIndex = 0;
            string firstTerm = "";
            string secondTerm = "";
            string indicators = "";
            string[] comparisonIndicators = new string[] { "الفرق بين", "الاختلاف بين"};
            string[] sentencesDelimeters = new string[] { ";", ".", "؟", "؛", "?", "!", "\r\n",
"\n\n"};
            StringBuilder term = new StringBuilder();
            StringBuilder definition = new StringBuilder();
            Dictionary<string, string> definAnswer = new Dictionary<string, string>();
            var sentences = ReadFromFileALL().Split(sentencesDelimeters,
StringSplitOptions.None).ToList();
            foreach (var sent in sentences)
            {
                indicators = "";
                term.Clear();
                definition.Clear();
                degree = 0;
                firstTerm = "";
                secondTerm = "";
                index = 0;
                sentIndex = 0;
                wordIndex = 0;
                dW = RemoveDiacritics(sent);
                indicators = comparisonIndicators.Where(x =>
dW.Contains(x)).FirstOrDefault();
                try
                {
                    if (!String.IsNullOrEmpty(indicators))
                    {
                        degree++;// increment matching degree
                        var words = SplitTextIntoWords(sent);
                        wordIndex = words.IndexOf(indicators.Split(new string[]{" "},
StringSplitOptions.RemoveEmptyEntries)[1]);
                        for (int j =  wordIndex + 1; j < words.Count(); j++)
                        {
                            if (!words[j].StartsWith("و"))
                            {
                                firstTerm += words[j] + " ";
                            }
                            else
                            {
                                index = words.LastIndexOf(words[j]);
                                j = words.Count();
                            }
                        }
                        for (int j = index ; j < words.Count(); j++)
                        {
                            secondTerm += words[j] + " ";
                        }
                        secondTerm = secondTerm.StartsWith("و") ? secondTerm.Substring(1,
secondTerm.Length - 1) : secondTerm;
                        var a = SplitTextIntoWords(firstTerm);
                        var b = SplitTextIntoWords(secondTerm);
                        int nounsInFirstTerm = 0;
                        int nounsInSecondTerm = 0;
                        foreach (var w in a)
                        {
                            if (w.StartsWith("ال"))
                            {
                                nounsInFirstTerm++;
                            }
                        }
                        foreach (var w in b)
                        {
                            if (w.StartsWith("ال"))
                            {
                                nounsInSecondTerm++;
                            }
                        }
                        if (nounsInFirstTerm > 0 && nounsInSecondTerm > 0)
                            degree++;

                        if (degree > 1)
                        {
                            term.Append(sent);
                        }
                        sentIndex = sentences.IndexOf(sent) + 1;
                        while (sentIndex < sentences.Count())
                        {
                            string nextSent = sentences[sentIndex];
                            if (!String.IsNullOrEmpty(nextSent) &&
(nextSent.Contains(firstTerm)))
                            {
                                nextSent = nextSent.Substring(nextSent.IndexOf(firstTerm),
nextSent.Length - nextSent.IndexOf(firstTerm));
                                definition.Append(nextSent + ". ");
                            }//if not empty

                            else if (!String.IsNullOrEmpty(nextSent) &&
(nextSent.Contains(secondTerm)))
                            {
                                nextSent = nextSent.Substring(nextSent.IndexOf(secondTerm),
nextSent.Length - nextSent.IndexOf(secondTerm));
                                definition.Append(Environment.NewLine);
                                definition.Append(Environment.NewLine);
                                definition.Append("أما "+ nextSent + ". ");
                            }
                            sentIndex++;
                        }//while not reach last of list
                        if (degree >= 1 && definition.Length > 0)
                        {
                            if (!definAnswer.ContainsKey(term.ToString()))
                            {
                                definAnswer.Add(term.ToString(), definition.ToString());
                            }
                        }
                    }//sent contains indicators
                }//try
                catch (Exception ex) { result = ex.Message; }
            }//for each sent
            return definAnswer;
        }
```

// Extract verses of the Holy Quran

```
  public Dictionary<string, string> ExtractHollyQuran()
        {
            string dW = "";
            string result = "";
            List<string> stamnts = new List<string>();
            Dictionary<string, string> termAnswer = new Dictionary<string, string>();
            string[] wordsDelimiters = new string[] {"|", "\\", "/", ".", ";", ",", "؛", "?", "!",
"(", ")", "،", "[", "]", "{", "}", "\"", "\'", "\t", "\n", "\r\n", "\n\n" };
            try
            {
                using (LOB_DBEntities db = new LOB_DBEntities())
                {
                    var symboles = db.LOB_Symbols.Select(x => x.symbol).ToList();
                    wordsDelimiters = wordsDelimiters.Concat(symboles).ToArray();
                    var HollyQuran = db.LOB_HollyQuran_Verses.ToList();
                    var text = ReadFromFileALL();
                    stamnts = text.Split(wordsDelimiters,
StringSplitOptions.RemoveEmptyEntries).Where(x =>
!String.IsNullOrWhiteSpace(x) && !String.IsNullOrEmpty(x)).ToList();
```

```
                    var verses = HollyQuran.Where(x => stamnts.Any(y =>
RemoveDiacritics(y.Trim()).Equals(RemoveDiacritics(x.verse.Trim())))).Distinct().T
oList();
                    foreach (var v in verses)
                    {
                        if (!termAnswer.ContainsKey(v.surah_name.Trim() + " / رقم " +
v.verse_no))
                        {
                            termAnswer.Add(v.surah_name.Trim() + " / رقم " + v.verse_no,
v.verse );
                        }
                        else { continue; }
                    }//foreach
                }//using DB
            }//try
            catch (Exception ex) { result = ex.Message; }
            return termAnswer;
        }
```

// Extracting the cause and effect

```
    public Dictionary<string, string> ExtractCauseEffect()
        {
            LOB_DBEntities db = new LOB_DBEntities();
            string result = "";
            string dW = "";
            int degree = 0;
            int sentIndex = 0;
            int indicatorsStartIndex = 0;
            int indicatorsEndIndex = 0;
            string indicators = "";
            string indicators2 = "";
            string mediator = "";
            string[] causeEffectIndicators = new string[] { "أدى الى", "بسبب ذلك", "الذلك", "لذلك "
"لهذا السبب", "سبّب", "نتج عنه", "نتيجة ل", "السبب"};
            string[] effectCauseIndicators = new string[] { "سبب ذلك", "بسبب", "نتيجة ل", "
"ذلك بسبب", "ناتج عن", "سببها" , "سببه"", ""};
            string[] sentencesDelimeters = new string[] { ";", ".", "؟" ,"؛", "?", "!", "\r\n",
"\n\n", "،", "," };
            string[] splitIndicatorsToList = new string[] { };
            StringBuilder firstTerm = new StringBuilder();
            StringBuilder secondTerm = new StringBuilder();
            Dictionary<string, string> termAnswer = new Dictionary<string, string>();
            var sentences = ReadFromFileALL().Split(sentencesDelimeters,
StringSplitOptions.None).ToList();
            foreach (var sent in sentences)
            {
                indicators = "";
                indicators2 = "";
                mediator = "";
                firstTerm.Clear();
                secondTerm.Clear();
                degree = 0;
                sentIndex = 0;
                indicatorsStartIndex = 0;
        indicatorsEndIndex = 0;
        splitIndicatorsToList = null;
        dW = RemoveDiacritics(sent);
        indicators = causeEffectIndicators.Where(x =>
dW.Contains(x)).FirstOrDefault();
        indicators2 = effectCauseIndicators.Where(x =>
dW.Contains(x)).FirstOrDefault();
        try
        {
            //cause-effect
            if (!String.IsNullOrEmpty(indicators) ||
!String.IsNullOrEmpty(indicators2))
            {
                degree++;// increment matching degree
                var words = SplitTextIntoWords(sent); // split sentences into words

                //split the indicator and get it's boundaries
                if (!String.IsNullOrEmpty(indicators))
                {
                    splitIndicatorsToList = indicators.Split(new string[] { " " },
StringSplitOptions.RemoveEmptyEntries);
                }
                else if (!String.IsNullOrEmpty(indicators2))
                {
                    splitIndicatorsToList = indicators2.Split(new string[] { " " },
StringSplitOptions.RemoveEmptyEntries);
                }
                if (splitIndicatorsToList.Count() > 0)
                {
                    indicatorsStartIndex = words.IndexOf(splitIndicatorsToList[0]);
                    indicatorsEndIndex =
words.IndexOf(splitIndicatorsToList[splitIndicatorsToList.Length - 1]);
                }

                //forming the first term
                for (int j = 0; j < indicatorsStartIndex; j++)
                {
                    //exclude al-hurruf
                    if (words[j].Length > 1)
                    {
                        firstTerm.Append(words[j] + " ");
                    }//if
                }//for

                //forming the second term
                for (int j = indicatorsEndIndex + 1; j < words.Count(); j++)
                {
                    // exclude al-hurruf
                    if (words[j].Length > 1)
                    {
                        secondTerm.Append(words[j] + " ");
                    }//if
                }//for

                sentIndex = sentences.IndexOf(sent);
                //add the prev sentence as first term if the current sentence not the first
                if (firstTerm == null && sentIndex > 0)
```

```
                        {
                            firstTerm.Append(sentences[sentIndex - 1]);
                        }//if

                        //!! not important condition
                        //add the next sentence as second term if the current sentence not the
last
                        if (secondTerm == null && sentIndex < sentences.Count() - 1)
                        {
                            secondTerm.Append(sentences[sentIndex + 1]);
                        }//if

                        //inverting the terms in case of effect-cause
                        if (!String.IsNullOrEmpty(indicators2) &&
String.IsNullOrEmpty(indicators))
                        {
                            mediator = firstTerm.ToString();
                            firstTerm.Clear();
                            firstTerm.Append(secondTerm);
                            secondTerm.Clear();
                            secondTerm.Append(mediator);
                        }
                        //add the terms to the results dictionary if the degree of matching is
achieved
                        if (degree >= 1 && firstTerm.Length > 0 && secondTerm.Length > 0)
                        {
                            if (!termAnswer.ContainsKey(firstTerm.ToString()))
                            {
                                termAnswer.Add("السبب : " + firstTerm.ToString().Trim(), "النتيجة :
" + secondTerm.ToString().Trim());
                            }
                        }//if
                    }//sent contains indicators
                }//try
                catch (Exception ex) { result = ex.Message; }
            }//for each sent
            return termAnswer;
        }
```

F. Results of the experiment

……….