Alqalam Journal of Medical and Applied Sciences. 2026;9(5):1332-1342
https://doi.org/10.54361/ajmas.269529

Original article

Arabic Domain Modeling: A Rule-based Tool for Extracting Domain Models from User Requirements using ANLP

Amina Mansouri*, Mohammed Hagel

Department of Software Engineering, Faculty of Information Technology, University of Benghazi, Benghazi, Libya
Corresponding Email. *amina.mansouri@uob.edu.ly*

Abstract
A domain model, also known as a conceptual model, is a crucial step in the transition from natural language requirements to precise specifications. It helps identify the main points of a problem in terms of the flow of real-world objects and their relationships. Domain modeling is also considered one of the best methods for requirements analysis. However, building a domain model manually for large systems is a challenging task. There are several ways to assist engineers with this task, such as using Natural Language Processing (NLP) to automatically extract candidate domain model elements. However, these methods are only applicable to the English language. Building a domain model in the Arabic language is particularly challenging due to the different concepts of the Arabic language compared to English. To assist analysts who use Arabic to document their work, a new approach for developing a tool that automatically extracts a domain model has been proposed. This is achieved by creating a set of rules extracted from the rules of the Arabic language and then reviewing them with language experts. Additionally, rules for writing requirements are imposed in accordance with recommended international standards. The tool showed effective performance in a case study presented in this research, with a performance rate of 87.9%.
Keywords: Arabic NLP, Domain Modeling, User Requirements, Arabic Language Processing, Part of Speech.

Introduction

Requirement analysis is a crucial stage in the Software Development Life Cycle (SDLC). It is considered the foundation for the success of any project as it helps determine the needs and conditions to meet for a new or altered product or project [1]. To strengthen its outputs due to the large number of documents used in it, this stage has been supported with several tools and models. One of these models is the Domain Model (DM), which is used to identify objects that can participate in the area of the problem [2]. However, what if this task is done automatically? It will undoubtedly make the analysis stage more solid and powerful. On the other hand, we will face many challenges in addressing the language used in writing the requirements, especially since we are targeting the Arabic requirements in this study. The task of analyzing software systems is crucial for any software engineer, as it forms the foundation upon which the entire system is built. Therefore, having tools that support analysts' work is essential and can significantly enhance the quality of software products.

During object-oriented analysis, domain modeling is used to break down the domain into real-world concepts or objects, which can be either ideas or actual objects. This process helps us define and organize the concepts and ideas related to the domain. Many previous attempts have been made to manipulate Arabic language processing using various methods. However, the treatment of the Arabic language was limited to analyzing feelings based on text, identifying geographical areas based on words used, and explaining the meaning of words to facilitate reading, such as translation. There have been some efforts to use ANLP to create Unified Modeling Language (UML) diagrams, such as use cases [3, 2-8]. This study proposes an automated method to extract the Domain Model (DM) from client requirements in Arabic. The method involves creating a set of rules based on the Arabic language and the writing standards. This will ensure greater accuracy and limit errors. A tool will be developed to automatically extract the DM based on the rules and natural language processing techniques. Additionally, a database will be created to store keywords used in the scenarios, which will increase Arabic language resources available for Machine Learning (ML) to perform software analysis tasks.

The domain model serves as a crucial step in transitioning natural-language requirements into precise specifications. However, building a domain model manually for large systems can be a laborious task. To assist engineers, various approaches have been developed to automatically extract candidate domain model elements using NLP. Despite the existing work, important facets remain underexplored. Firstly, there is limited empirical evidence about the usefulness of existing extraction rules (heuristics) when applied in industrial settings. Secondly, existing extraction rules do not adequately exploit the natural-language dependencies detected by modern NLP technologies. Lastly, an important class of rules developed by the information retrieval community for information extraction remains unutilized for building domain models. To address these limitations, this study developed a domain model extractor. The extractor brings together existing extraction rules from the software engineering literature for English grammar, extends these rules with complementary rules from the information retrieval literature, and proposes new rules to better utilize the results obtained from modern NLP [4].

Automated software engineering has attracted a large amount of research effort. The use of object-oriented methods for software system development has made it necessary to develop approaches that automate the construction of different UML models in a semi-automated manner from textual user requirements. UML

Copyright *Author (s) 2025. Distributed under Creative Commons CC-BY 4.0*
Received*: 14-03-2025 -* ***Accepted****: 11-05-2025 -* ***Published****: 18-05-2025*

use case models represent an essential artifact that provides a perspective on the system under analysis or development. A set of rules was defined for each element in the use case to be used, and a natural language processing tool was used to parse different statements of the user requirements written in Arabic to obtain lists of nouns, noun phrases, verbs, verb phrases, etc., that aid in finding potential actors and use cases.[3] Automated software engineering is one of the most important current research problems, especially when it comes to requirements analysis and modelling. The main advantage of this automation process is to improve the quality and productivity of software development. This study presented a semi-automated approach for constructing the activity diagrams from Arabic user requirements using the MADA+TOKAN parser. MADA+TOKAN used a set of tags to describe the Arabic words in the statements. Each word has a tag, and every tag has a special meaning. A set of rules is defined for each element in the activity diagram to be used during the generation of the diagram [11].

The sequence diagram is a commonly used UML model during the analysis phase of software system development. However, as creating such diagrams is typically a manual process, the addition of automated or semi-automated support would be beneficial and practical. This paper proposed a semi-automated approach for generating sequence diagrams from user requirements written in Arabic. Our study outlines a method for parsing user Arabic requirements using a natural language processing tool to generate part-of-speech tags. The resulting sequence diagram can then be created using sequence diagram drawing tools and represented using XMI [1]. Previous studies have shown that there is a gap in research when it comes to the domain model and the Arabic language. No study has been conducted to address this issue. Additionally, it has been observed that other models, such as the use case diagram, sequence, and activity, have been used to provide solutions for the Arabic language, but these models have not developed tools and instead presented semi-automatic methodologies.

Domain modeling is an essential step in software development that helps to identify the different elements that make up the target system and their relationships. It simplifies the transition from natural language requirements to precise specifications. However, constructing domain models manually is a time-consuming task, especially in large-scale systems where numerous target documents are required. Most of the existing methods for building domain models rely on rules that extract domain model elements, but these rules are defined only for English grammar rules [4], and cannot be directly applied to Arabic language requirements documents. Arabic readability research faces an obstacle due to the lack of sufficiently large datasets for annotators to provide labels with proper readability assessments. To test new tools, it is necessary to construct a dataset that can serve as a gold standard [17]. This study aims to develop a tool that helps software engineers extract DM from requirements written in Arabic. By doing so, it will improve the process of analyzing software systems and raise the quality of the software development process with minimal time and effort.

The Proposed Approach

There are two main processes involved in the approach: the theoretical process and the practical process. The theoretical process explains how to define rules for writing texts in Arabic, recommended standards for writing software requirements, and how to verify the ability to apply these rules practically. The practical process explains how a tool will be prepared to analyze the text, find objects and their properties, and understand relationships between them. Additionally, it demonstrates how the domain model for the entrance scenario will be automatically generated. In Figure 1, you can see a conceptual overview of the proposed Approach.

Copyright *Author (s) 2025. Distributed under Creative Commons CC-BY 4.0*
Received*: 14-03-2025 -* ***Accepted****: 11-05-2025 -* ***Published****: 18-05-2025*

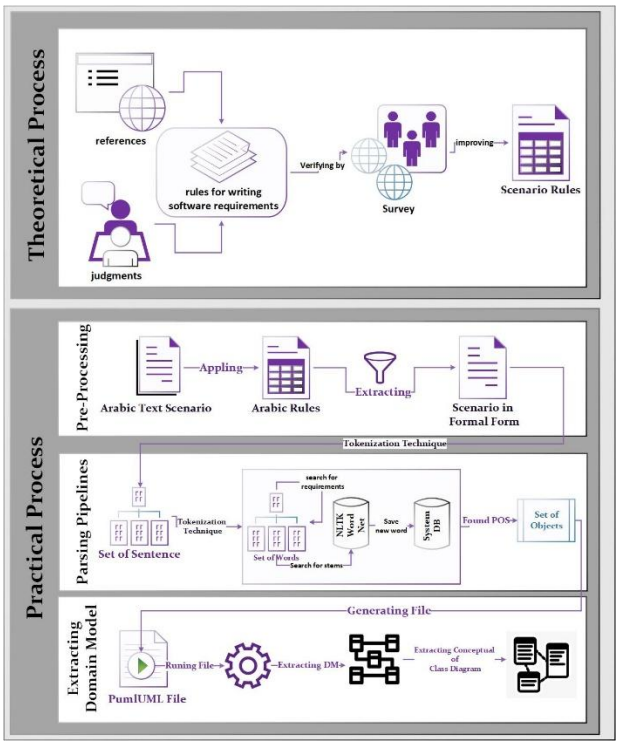


Figure 1. Conceptual Overview of the proposed Approach.

The approach consists of two main processes: the theoretical process and the practical process. In a theoretical process, it was explained how to define the rules for writing texts in Arabic and the recommended standards for writing software requirements, as well as how the ability to apply these rules on the ground will be verified. In a practical process explained how an algorithm will be prepared to analyze the text and find objects, their properties, and the relationships between them, and demonstrated how the domain model for the entrance scenario will be generated.

a) Theoretical Process

The purpose of this process is to establish the Arabic rules that should be applied when writing software requirements and extracting objects and their relationships based on the Arabic rules and recommended guidelines. This set of rules will assist the developed tool in automatically capturing the domain model from the provided Arabic texts using ANLP. The research used a Rule-based NLP approach, which utilizes extensive libraries of human language rules to achieve more accurate tagging. A rule-based NLP system follows these rules to classify the language it is analyzing.[15]

- *Data Collecting:* The rules for writing Arabic script were gathered by examining a collection of online e-books that outline the restrictions on writing sentences in Arabic. Table 1 shows the rules used to enforce the guidelines for writing software system scenarios. In Arabic writing, punctuation marks play a crucial role in identifying places of separation, stopping, and starting. Various references emphasize the importance of using these marks to facilitate comprehension for both the writer and reader [13, 12, 6]. Additionally, we reviewed a recommended guide for describing software systems, which highlights several critical characteristics that must be present in the requirements when documenting them, such as unambiguous, complete, verifiable, consistent, modifiable, and traceable [7].

Table 1. Rules used to enforce the rules for writing software system scenarios.

Rule No.	Description	Example
S1	The end of the sentence by a full stop (.)	مثل عند انتهاء التسجيل بالمدرسة يقوم موظف الشؤون الادارية بإعداد كشف بأسماء التلاميذ حسب الفصول الدراسية.
S2	The properties of the object declared using by Punctuation (:)	مثل يقوم الموظف بتسجيل بيانات التلميذ الجديد و التي تشمل : اسم الطالب الرباعي ، سنة الميلاد ، اسم الام الثلاثي ، رقم الهاتف ، العام الدراسي ، المرحلة الدراسية.
S3	The properties of the object splitting by using the punctuation comma (،)	مثل يقوم الموظف بتسجيل بيانات التلميذ الجديد و التي تشمل : اسم الطالب الرباعي ، سنة الميلاد ، اسم الام الثلاثي ، رقم الهاتف ، العام الدراسي ، المرحلة الدراسية.

Copyright *Author (s) 2025. Distributed under Creative Commons CC-BY 4.0*
Received*: 14-03-2025 -* ***Accepted****: 11-05-2025 -* ***Published****: 18-05-2025*

https://doi.org/10.54361/ajmas.269529

Rule No.	Description	Example
S4	The existence of a variable is declared by the (shall's statement), where requirements must begin with the word shall or its synonym.	مثل **يجب** على النظام ان يسمح للمستخدم بتسجيل دخوله من خلال : اسم المستخدم ، كلمة المرور.

The rules S1, S2, S3, and S4 display how to write software system scenarios in the correct form. For a more detailed explanation of Rule S4 in Table I, consider the following example (Figure 2 and Figure 3), which shows one of the ways requirements can be written, such as declarative "يـجـب" statements. Many characteristics have been imposed by international standards to define the form of the requirement, including that it begins with the word "يجب" or an equivalent meaning for it like "يسمح ، يمكن ، يوفر ، يقوم". So these rules facilitate the extraction of the domain model from Arabic requirements.
The word "يجب" or an equivalent meaning for it is a sufficient statement for the client's need for this function. Imposing rules helps word processing algorithms perform their functions more accurately. Therefore, these rules were specified to help the research algorithm perform its work, as the algorithm discovered that there was a requirement for the entered Arabic text, as shown in Figure 2, to be included in the analysis and search process for the parts of speech and extracting the objects, as shown in Figure 3, that can be involved in building the domain model for this sentence.

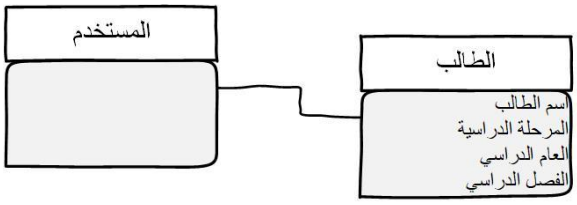


Figure 2. Capturing the domain model for the statement (fig. 3).

Figure 3. Shows example requirements statement with "shall" for rule no. S4.

The algorithm looks for the keyword as "يقوم", then searches for (المفعول به و الفاعل), where the first noun which is "المسـتخدم" represents the first object, and the second noun, which is "الطالب" represents the first object. The word "بتسجيل" represents the relationship between the subject (الفاعل) and the object (المفعول به).
The grammatical meaning of the word "المستخدم" in the previous sentence is "subject, فاعل " the word "الطالب" is "object, مفعول به " and the word "بتسجيل"" is "اسم مصدر".
There is a book available along with some summaries on the Arabic language for the first intermediate grade. These resources explain the most important spelling rules in the Arabic language, which can help in improving writing skills. In Figure 4, you can see an explanation of the grammatical classification that was used by the tool developed for this research to extract POS from scenarios that were collected for some software systems.[9]
Speech in the Arabic language is classified into (اسم ، فعل ، حرف) Each part depends on a set of standards when writing it, for example, the name (الاسم) must begin with "الـ" and the letter (الحرف) are taken care of in a group of words such as "في ، الى". As for the verb (الفعل), there are three types of it, each of which imposes a specific method of writing: the past ending with "تاء الفاعل", the present begin with "سـ او سوف" and the command begin with "أ".[9]

Copyright *Author (s) 2025. Distributed under Creative Commons CC-BY 4.0*
Received*: 14-03-2025 -* ***Accepted****: 11-05-2025 -* ***Published****: 18-05-2025*

https://doi.org/10.54361/ajmas.269529

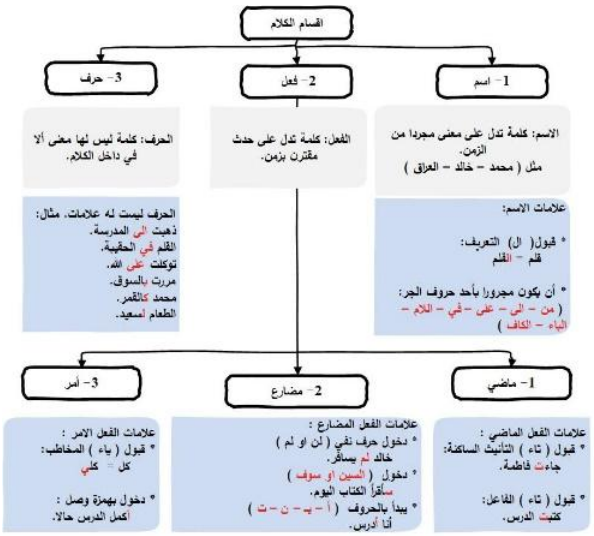


Figure 4. Show the summary of Arabic grammar [9]

The study also relied on another source that clarifies the various types of sources in the Arabic language, their definitions, uses, and rules. Figure 5, presented in this source, explains the standards and types of sources in the Arabic language.[10]

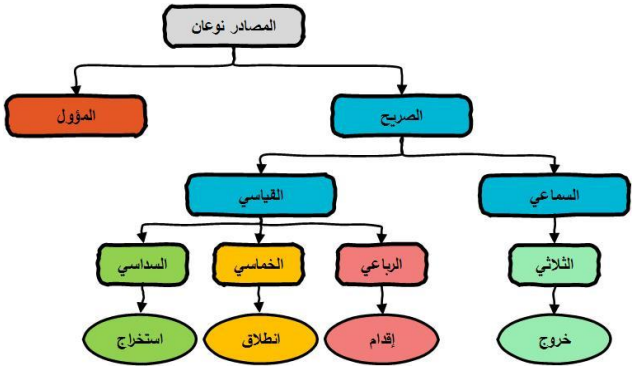


Figure 5. Show the types of sources in the Arabic language.[10]

The source of its meaning is a specific event that is not linked to time. It is divided into two main types as follows: In terms of the form (سماعي - قياسي) in terms of the number of letters (المصدر ثلاثي – رباعي – خماسي – سداسي). (المصادر الغير ثلاثية) is not governed by a rule, as its sources are auditory, not standard. As for (الثلاثي), there are rules governing their writing. Through analyzing a set of scenarios of software systems, it was noticed that the (المصـدر الرباعي) was used frequently, and its presence indicates the existence of a relationship between objects. The rule of (المصدر الرباعي) depends on the presence of the letter (ياء) before the last letter in the word, such as: "تسجيل" the source of this word is "سجل".[10]
The rules shown in Table 2 were used to extract the grammatical classification of nouns and verbs in Arabic language references, to find objects and relationships from nouns and keywords from verbs.

Table 2. Rules used to enforce the rules for writing software system scenarios.

Rule No.	Description	Example
O1	The object noun accepts (الـ) definition.	مثل : يعمل **الموظف** على اضافة بيانات المعلمين الجدد.
O2	Being is a noun preceded by a preposition (**من ، الى ، على ، في ، اللام ، الباء ، الكاف**،).	مثل: يسمح **للموظف** بإضافة بيانات جدول الحصص في حالة امتلاكه لصلاحية مدير.
O3	The name may be a connected pronoun.	مثل: يستطيع **الموظف** ادخال بيانات رواتب الموظفين بالمدرسة ، و يمكنـه ايضا معالجة بيانات السلف لإي موظف.

Copyright *Author (s) 2025. Distributed under Creative Commons CC-BY 4.0*
Received: *14-03-2025* - ***Accepted***: *11-05-2025* - ***Published***: *18-05-2025*

Rule No.	Description	Example
K1	The relation comes in the form of a past tense that accepts the (**تاء الفاعل**) of a participle.	مثل: **اصدرت** اعلاناً عن امكانية قبول تلاميذ جدد.
K2	The relation comes in the form of a present tense may be preceded by a negative letter (**لن او لم**).	مثل: لن **يسمح** للمستخدم من تحديد الفصل الدراسي لأي تلميذ الا بعد ترحيله.
K3	The relation comes in the form of a present tense may be preceded by a letter (**السين او سوف**).	مثل: **سيقوم** الموظف بطباعة قائمة بأسماء التلاميذ لكل مرحلة.
K4	The relation comes in the form of a present tense may begin with a letter (**أ – نـــ – يـــ – تـــ**).	مثل: يقوم الموظف **بإصدار** ايصال مالي بالقيمة المستلمة من ولي الامر.
R1	If the verb is preceded by a preposition, the presence of the letter (**ياء**) before the last letter in the word this noun may represent the (**مصدر رباعي**) of the verb.	مثل: يقوم الموظف **بتسجيل** بيانات الطالب. **"تسجيل"** فعل غير مقترن بزمن (اسم مصدر لفعل رباعي سجّل).
R2	If the verb is preceded by a preposition, the presence of the letter (**الالف**) before the last letter in the word this noun may represent (**مصدر رباعي**).	مثل: يقوم الموظف **بإصدار** ايصال مالي بالقيمة المستلمة من ولي الامر. **"إصدار"** فعل غير مقترن بزمن (اسم مصدر لفعل رباعي أصدَرَ).
R3	If the verb is preceded by a preposition, the presence of the letter (**الالف**) at the beginning of the word may represent this noun (**مصدر رباعي او خماسي**).	مثل: يقوم الموظف **بإصدار** ايصال مالي بالقيمة المستلمة من ولي الامر. **"إصدار"** فعل غير مقترن بزمن (اسم مصدر لفعل رباعي أصدَرَ).
R4	If the verb is preceded by a preposition, the presence of the letter (**الميم**) at the beginning of the word may represent this noun (**مصدر ميمي**).	مثل: يسمح للموظف من **معالجة** بيانات السلف بعد تحديد الراتب لكل موظف. **"معالجة"** مصدر يبدأ بميم مضمومة زائدة، ويدل على حدث ما.

The rules O1, O2, and O3 display how to delimit objects. The rules K1, K2, K3, and K4 display how to delimit verbs that may be pointed to key words. The rules R1, R2, R3, and R4 display the name of the relationship between the objects.

In Figure 6 shows the grammatical classification of each word in the sentence (يجب أن يكون لدى الموظف صلاحية تسجيل بيانات الطلبة الجدد) which was output by the research tool that was developed in accordance with the rules in the preceding table.

Figure 6. Shows grammatical classification of each word.

The Figure show set of steps: step 1, the developed tool looks for the first noun that constitutes the first object in this sentence, which is (الموظف) after finding the keyword for the presence of a requirement, which is the verb (يسمح). In step 2, it looks for the source name that joined two objects, which is (تسجيل). In step 3, it looks for the second noun that constitutes the second object in the sentence, which is (الطلبة). An increase in clarification of the true grammatical meaning of the Arabic grammar, which was represented in the previous figure in the symbols (K1, O1, R1, O2), where the sentence structure was a verbal sentence, which are the sentences that begin with a verb in the Arabic language, means that the tool that was developed in this research, is interested in finding the noun and the verb based of their grammar parsing, which is represented in the form of the previous one as follows: K1 represents the verb and serves as an indicator word to the tool

Copyright *Author (s) 2025. Distributed under Creative Commons CC-BY 4.0*
Received*: 14-03-2025 -* ***Accepted****: 11-05-2025 -* ***Published****: 18-05-2025*

for the existence of a requirement, while O1 represents the subject that represents the noun that may be the first object in the system, R1 represents the event that the subject performs, and it represents the name of the relationship that connects the first object with the second, and O2 represents the object that represents the second object in the system.

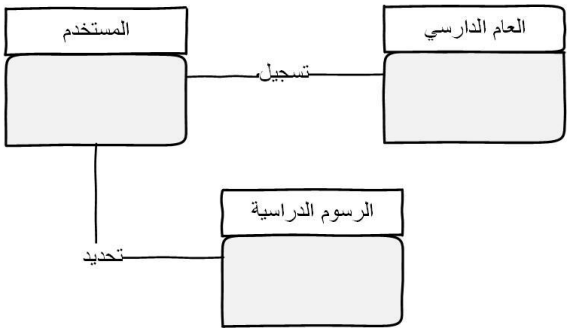


Figure 7. Capturing domain model for the statement (fig. 8).

المتطلب1: يقوم المستخدم بتسجيل بيانات الطالب الجديد. ثم يمكنه النظام من تحديد الرسوم الدراسية لكل مرحلة.

Figure 8. Shows example requirements statement with connected pronoun for rule no. O3.

In Figure 7, there is a relationship between the words "المستخدم" and "الرسوم الدراسية" through the pronoun "ـــــه" in the word "يمكنه". Based on this dependency the relationship in fig. Figure 8 can be derived.
The algorithm looks for the keyword as "يقوم", then it looks for first object then for relationship name and second object, but in the previous example Figure 7, the second part of a sentence was only presented relationship name and second object, which is "تحديد ، الرسوم الدراسية" without first object, this is a representation of the relation of connected pronoun "ـــــــــه", which means that is related to the subject mentioned at the beginning of the sentence "المستخدم", so the algorithm calls the name of the last object which is "المستخدم" that was mentioned to represent it in the relationship, as shown in Figure 8.

- *Data Verifying:* it is recommended to create a domain model either before or simultaneously with documenting the requirements. However, due to time and resource constraints, this may not always be feasible. Building a domain model that aligns with a given set of requirements requires the engineers to examine the requirements carefully and ensure that all the relevant concepts and relationships are included in the model. This can be a tedious task, especially for large applications with requirements spanning over tens or hundreds of pages. To address this challenge, automated assistance for constructing domain models based on natural language requirements is crucial.

To this end, a set of rules was compiled and restricted to facilitate the extraction of the domain model and to write software system scenarios. These rules were reviewed by Arabic language experts to validate them, and a survey was conducted to gather feedback from software engineers specializing in software development about the proposed tool's imposed rules. The feedback was also used to improve the suggested rules.
The number of participants in the survey was 60, with ‘48 females and 12 males’ of different ages, as their years of graduation ranged from 1988 to 2023. The participants were from different disciplines (39 from software engineering, 12 from computer science, 6 from information systems, 1 from networks and communications, and 2 from programming hobbyists).

b) Practical Process

The aim of this process is to create a tool that can extract the domain model from an Arabic requirements scenario. The collected rules will be applied to the entered text, then the objects, their properties, and the relationships between them will be extracted, and finally the domain model will be drawn and a conceptual model of the class diagram will be extracted.

- *Pre-Processing*: the following steps will be carried out manually to ensure that the resulting scenario is compliant with the rules of the Arabic language. This scenario will be relied upon in the subsequent stages of the tool's operation. It is important to note that the accuracy of the tool's output is dependent on the degree to which the scenario adheres to the imposed rules. If the scenario is in compliance with the rules, then the tool will accurately extract the domain model.
- *Parsing Pipeline*: the tool proposed in this study aims to analyze Arabic text sentence by sentence to generate a graphical domain model. Figure 9 illustrates the parsing pipeline for syntactic

Copyright *Author (s) 2025. Distributed under Creative Commons CC-BY 4.0*
Received*: 14-03-2025 -* ***Accepted****: 11-05-2025 -* ***Published****: 18-05-2025*

parsing. To analyze the entered text, the ANLP process employs a set of components that participate in the syntactic analysis pipeline. Each of these components has its own swim lane, which is represented horizontally. This stage comprises four swim lanes for the following components: NLTK, Set-Lists, Checker, and PlantUMl. These components were developed to work together to extract the domain model for the entered scenario, and their responsibilities are clearly defined within the tool.

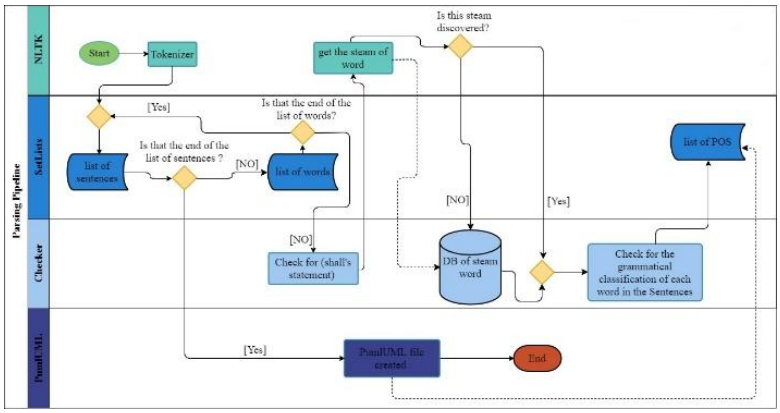
- 

Figure 9. Shows the parsing pipeline process.

The following steps detail the parsing pipeline process by identifying each responsible component's name and task:

1) **Component name: NLTK / Task name (Tokenization):** in a NLTK component, the entered text will be split using the tokenization technology.
2) **Component name: Set-Lists / Task name (Storage sentences and words):** in a set-lists component, a special set will be prepared to store the sentences. After completing each item in the list (i.e each sentence) will be divided and stored into a list of words.
3) **Component name: Checker / Task name (Checking for POS):** in a checker component, the tool worked with the rules in Table II regarding the rules of objects and the relationship between them. Look at Figure 10, which displays an example of a checker component.

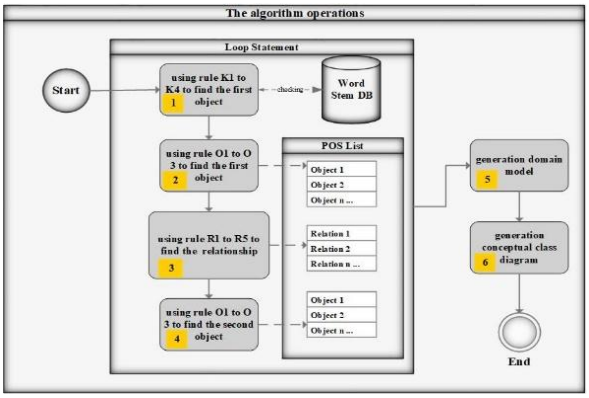


Figure 10. The checker's operations.

An example of a checker's operations will be applied with the sentence (يسمح للموظف بتسجيل بيانات الطلبة الجدد), (1)the tool will work to search for the sentence that may be a software requirement by applying the rules of K1 to K2 to each word in the sentence, until the verb that represents the word synonymous with the word 'يجب' is found, in this example, the word (يسمح) applies rule no. K4. (2)Applied the rules O1 to O3, searching to find the subject that will represent the first object in this example, the word (للموظف) applies rule no. O2. (3)Searching for the source name, which will result in a relationship between the expected objects in the sentence, in this example, the word (بتسجيل) applies rule no. R1. (4) Apply the rules O1 to O3 to search for the object that will represent the second object in this example, the word (الطلبة) applies rule no. O1.
If a relationship between the objects in the past sentence is discovered, the NLTK component will obtain the word stem representing the existence of the requirement discovered in the first step, which is (يسمح) and its stem was (سمح). The stem of the word (يسمح) will be checked in the database dedicated to stem words; if the word doesn't find a match, it will be kept. These steps will be repeated with the next sentence until all sentences are complete.

4) **Component name: PlantUML / Task name Extracting Domain Model:** After the list of sentences is finished, a DM is extracted by code which is generated for UML drawings from the specified list of

Copyright *Author (s) 2025. Distributed under Creative Commons CC-BY 4.0*
Received*: 14-03-2025 -* ***Accepted****: 11-05-2025 -* ***Published****: 18-05-2025*

objects, their properties, and the relationships among them by the developed tool. The developed tool uses PlantUML UML notations to generate its code, which is then applied by the developed tool to extract the DM. This makes all operations take place in a single operating environment, so that it can be said that the developed tool works fully automatically. Where we noticed in all previous studies that the proposed solutions are done with the help of a set of tools that are independent from each other and are not integrated during the provision of solutions, here these solutions are called semi-automatic. The study is pleased to present a solution that is different from all previous works by developing a performance capable of working fully automatically. Table 3 shows PlantUML notations and their meaning.[5]

Table 3. Plantuml notations and their meaning

Notation	The Meaning
@startuml	Start point.
class	It's a keyword used to set the name of a class, (i.e: class **A)**
{ }	Used to set the properties of the class.
,	Used to separate the properties of the class.
--	Used to draw the relationship between the classes.
:	Used to add a label on the relation.
@enduml	End point.

The expected domain model schema for the Arabic requirements text is obtained after running the generated file when the tool finishes working; this file contains all the objects expected to participate in the problem domain, in addition; their properties and the relationships between them, this step is done by PlantUML tool, it is an open source tool that allows to quickly write: Sequence diagram, Using case diagram, Class diagram, Activity diagram, Component diagram, State diagram, Object diagram.[5]
A file PlantUML begins with a (@startuml) and ends with a (@enduml), the existence of an object is expressed through the word (class), its properties are expressed by limiting them between mark ({}), separating them is expressed with a sign (,), and relations are expressed through the sign (--) between the two objects. It is possible to add a label on the relation by using (:).[5]

- *Post-Processing*: In this study, a tool is developed to work automatically to present the final diagram for the interred scenario. It is extracted by the developed tool that works by using ANLP techniques and also it works to generate automatically the PlantUML code to draw the domain model in the generated file. For more illustration, Figure 11, shows an example of the domain model for one of the scenarios of the software systems extracted by the tool that is developed in this research using the PyCharm IDE version 2022.2.2.

The following is an example used to show the result of processes in the parsing pipeline stage:
يقوم الموظف بإضافة بيانات العام الدراسي الجديد وذلك بإدخال : عنوان العام الدراسي ، رمز العام الدراسي. سيتمكن ايضا من إضافة بيانات الطالب الجديد وذلك بإدخال : اسم الطالب ، سنة الميلاد ، الرقم الوطني أو رقم جواز السفر ، الصورة الشخصية. في حالة رغبة ولي الامر بسحب ملف الطالب يقوم الموظف بإدخال بيانات النقل للطالب في حالة انسحابه من المدرسة وذلك بإدخال : رقم الطالب ، اسم ولي الامر ، سبب الانسحاب ، تاريخ الانسحاب. بعد ذلك يقوم الموظف بإدخال بيانات الترحيل لطالب المستمر داخل المدرسة للعام الدراسي الجديد من خلال ادخال : اسم الطالب ، المرحلة الدراسية ، العام الدراسي. ثم يقوم بإضافة المؤهل العلمي للمعلم من خلال ادخال : اسم المؤهل ، رقم المؤهل. بعد ذلك يقوم بإضافة التخصص للمؤهل العلمي من خلال ادخال اسم المؤهل ، اسم التخصص ، رقم التخصص. ثم يمكنه بتحديد المؤهل لكل معلم يدرس بالمدرسة.

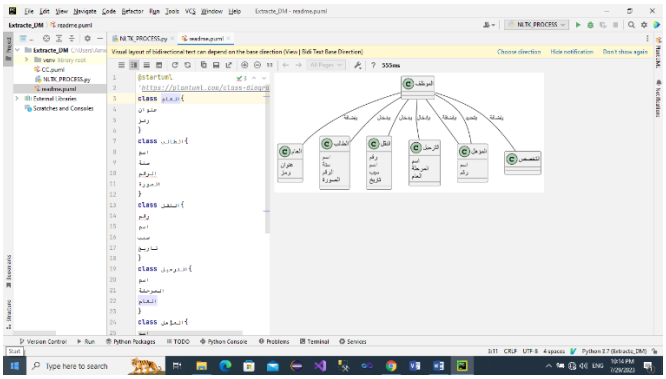


Figure 11. An example of the domain model extracted by the tool was developed using the PyCharm IDE.

Copyright *Author (s) 2025. Distributed under Creative Commons CC-BY 4.0*
Received*: 14-03-2025 -* ***Accepted****: 11-05-2025 -* ***Published****: 18-05-2025*

Discussion

Despite studies that suggest the importance of a domain model in software analysis, software engineers may not want to spend time deducing the objects that can participate in the problem domain. Instead, they might go directly to building the class diagram and database, or they may need to work in English to use available automatic processing tools. Therefore, it is necessary to conduct research that focuses on automatic processing of Arabic language texts to improve the quality of software development.

This study explains the importance of the domain model in the analysis stage and how it contributes to the success and understanding of software projects. Many previous studies have contributed to building the domain model. The domain model was automatically covered by analyzing texts in different languages such as English and Japanese. However, there hasn't been any study dealing with the domain model in Arabic. Although there are efforts to address this issue in other languages, none of them are in Arabic.[4]

The study discovered that Arabic text processing in software engineering has made significant progress, particularly in the area of semi-automatically extracting activity diagrams and use case diagrams from Arabic texts. However, the process did not rely solely on one tool to extract diagrams and process Arabic texts. Instead, multiple independent tools were used to achieve the final result. While previous studies have produced these diagrams semi-automatically, this study focused on developing a fully automatic tool to extract domain models from Arabic texts [3,11,1].

Based on the preceding analysis, the key outcomes of the study can be summarized as follows:

- Contribute to determining a set of rules for writing scripts for software systems in the Arabic language, based on the recommended rules for writing software requirements and grammatical rules in the Arabic language.
- Contribute to preparing a database of the origins of Arabic words for words similar to the shall statement (يجب), which indicates the existence of a requirement. This database will provide great support for the tools that work on processing Arabic texts and will also contribute significantly to supporting machine learning algorithms to train algorithms to discover the software requirements without having to parse the text.
- Contribute to determining a set of rules to extract the objects of the problem domain, their characteristics, and the relationship between them, based on the grammatical rules in the Arabic language.
- Contribute to developing the first tool to extract the domain model fully automatically, and this is based on the prepared approach, which was explained. The work of this tool was also evaluated by a group of software engineers through a questionnaire that was prepared and published on the Facebook App, and by case study for an administrative and financial system scenario, the results indicated that the performance rate was 87.9% after comparing the conceptual class that the tool envisioned and the real classes extracted by software engineers.

Conclusion

The study presented a compilation approach that contains two important processes (theoretical and practical). The theoretical process developed a plan to collect all data related to the grammatical rules in the Arabic language and the recommended standards for writing the software requirements. This process was also concerned with determining the appropriate mechanism for preparing the questionnaire and publishing it. The questionnaire was designed to assess the ability of software engineers to apply these rules during their real work, and then the practical process was launched to develop the tool dedicated to extracting the domain model from Arabic texts, where the rules used in extracting the domain model were defined in the theoretical process. These rules significantly contributed to the extraction of parts of speech (POS) from texts with the development environment (PyCharm Community Edition 2022.2.2) of the language, and the program language (Python) that was used to write code commands, the (NLTK) library integrated with this environment was also used, which is a huge library that includes a group of classes in everything related to processing Natural language, and finally the markup language (PumlUML) used in drawing UML diagrams was used, where the tool will generate the (PumlUML) code for the domin model of text that was entered in a separate file automatically to display the final graphic form of the classes expected to participate in the problem domain.

This study seeks to promote scientific research and provide solutions for Arab software engineers who use their language in software documentation. One of the main results from the evaluation of the tool's performance is the consensus of the Arab engineers participating in the questionnaire on the importance of this tool and their need for it in real life. In future work, the tool will be improved to accurately extract classes, their operations, and the type of relationship between them (class schema). It can also be done to generate code according to the language specified by the user for each class discovered with the set and get functions. In addition, we work to make the tool more intelligent by using the database that has been prepared.

Competing Interests

The authors declare that they have no competing interests.

Copyright *Author (s) 2025. Distributed under Creative Commons CC-BY 4.0*
Received*: 14-03-2025 -* ***Accepted****: 11-05-2025 -* ***Published****: 18-05-2025*

Authors' Contributions

Amina Ali Mansouri conducted the main research, designed the tool, and wrote the manuscript. Mohamed A. Hagal supervised the study, contributed to the tool design, and provided critical revisions. Both authors read and approved the final version of the manuscript.

Ethical and Informed Consent

This study did not involve human participants, personal data, or experiments requiring ethical approval. All data used in the study were either publicly available or simulated for research purposes.

Data Availability Statement

The datasets generated and analyzed during the current study are available from the corresponding author upon reasonable request.

References

1. Alami, N.; Al-Kabi, M.N.; Wahsheh, H.A. A semi-automated approach for generating sequence diagrams from Arabic user requirements using a natural language processing tool. In Proceedings of the 8th International Conference on Information Technology (ICIT), Amman, Jordan, 17–18 May 2017.
2. Albahli, S. Twitter sentiment analysis: An Arabic text mining approach based on COVID-19. Front. Public Health 2022, 10.
3. Arman, N.; Jabbarin, S. Generating use case models from Arabic user requirements in a semiautomated approach using a natural language processing tool. J. Intell. Syst. 2015, 24(2).
4. Arora, C.; Sabetzadeh, M.; Briand, L.; Zimmer, F. Extracting domain models from natural-language requirements. In Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems, Saint-Malo, France, 2–7 October 2016.
5. PlantUML Language Reference Guide. Available online: https://plantuml.com/guide (accessed on 31 March 2026).
6. ديوان العرب. علامات الترقيم في الكتابة العربية ومواضع استعمالها. Available online: http://www.diwanalarab.com (accessed on 31 March 2026).
7. IEEE. An American National Standard IEEE Guide to Software Requirements Specifications; IEEE: New York, NY, USA, 1984.
8. Larabi Marie-Sainte, S.; Guessoum, A.; Elkhlifi, A. Arabic natural language processing and machine learning-based systems. IEEE Access 2019, 7, 7011–7020.
9. مهدي، أ. قواعد اللغة العربية أول متوسط؛ إعداد الأستاذ أحمد العراقي؛ الأستاذ أحمد مهدي شلال عباس المهداوي، 2022. Available online: http://www.amsebehm2017.com/2020/12/blog-post_20.html (accessed on 31 March 2026).
10. مجلة محطات. انواع المصادر في اللغة العربية. Available online: http://www.mah6at.net (accessed on 31 March 2026).
11. Nassar, I.N.; Khamayseh, F.T. Constructing activity diagrams from Arabic user requirements using natural language processing tool. In Proceedings of the 6th International Conference on Information and Communication Systems (ICICS), Amman, Jordan, 7–9 April 2015.
12. قباوة، ف. علامات الترقيم في اللغة العربية؛ 2007.
13. الحقاني، ف.ر. علامات الترقيم وأصول الإملاء؛ دار الكتب العلمية: بيروت، لبنان، 2016.
14. Rasch, D. Review of Software Engineering by Pressman, R.S. Biometrical Journal 1991, 33(3), 378–378.
15. SentiSum. Machine learning vs rule-based NLP. Available online: https://www.sentisum.com/success-article/machine-learning-nlp (accessed on 31 March 2026).
16. Sperber, M. Review of Functional and Reactive Domain Modeling. J. Funct. Program. 2020, 30.
17. Wahdan, A.; Younes, M.; Al-Zoghby, H. Text classification of Arabic text: Deep learning in ANLP. In Advances in Intelligent Systems and Computing; Springer: Cham, Switzerland, 2021; pp. 95–103..

Copyright *Author (s) 2025. Distributed under Creative Commons CC-BY 4.0*
Received*: 14-03-2025 -* ***Accepted****: 11-05-2025 -* ***Published****: 18-05-2025*