International Journal of Information Technology and Language Studies (IJITLS)

A Framework for Building an Educational Question Answering System

Heba Samy[1], **Ehab E. Hassanein**[1], and **Khaled Shaalan**[2]
hbsamy@gmail.com; e.ezat@fci-cu.edu.eg; khaled.shaalan@buid.ac.ae

[1] Faculty of Computers & Information, Cairo University, Egypt
[2] Faculty of Engineering & IT, The British University in Dubai, UAE

Abstract. A question answering unit in any educational system can be very helpful and effective for enhancing the learning process. This paper introduces a design for the building blocks and workflow for an educational question answering system that can enhance the learning process. This system uses an Arabic ontology as a prototype to prove its efficiency. It depends mainly on natural language processing techniques that exist in the research area. A rule-based question classification method is created as well as a querying process depending on named entity recognition and pre-made templates.

Keywords: *Question Answering, Question Answering systems, Ontology, linked data, e-learning, online education, ontology-based Arabic Question Answering*

1. Introduction

"Asking good questions is a central process in learning and sometimes it's even more important than finding the answers, especially when the questions foster the learner's critical thinking. That's one can say that questioning is a basic skill for good education" (Betts, 1910, p. 55). Asking questions is also good for educators as it helps them to investigate the knowledge gaps of learners by digging into their thinking, evaluate their high-order thinking, and adding or adapt part of the rest of the learning topic so that it enables students to influence the curriculum under study and to get more motivated to learn (Chin et al., 2008). But despite its importance, not all types of people/ages can get the benefit. For example, for higher grades, researchers have found that in most educational environments whether it's offline or online, higher grades students tend to ask fewer questions than those of lower grades. That's maybe because they don't like to draw attention or their instructors are not welcoming more questions (Chin et al., 2008).

For this type of people, that are teenagers or shy people, an online educational question answering system can serve as a safe place for learners to ask questions without being noticed by their colleagues and give the instructor more time to check learners' questions, analyze, and answer them as well. This question answering system should be able to answer students instantly from an underlined database containing the curriculum understudy. Then the instructor would be able to check these questions and answers for more educational enrichment.

Question answering as a term in information retrieval is defined as the process of a computer answering a human question that is being asked in his/her natural language, depending on either an underlined structured database or a repository of text documents in natural language (Samy et al., 2019). In comparison to other Latin languages, Arabic is somehow a challenging language when building a question answering system. This is because of the lack of tools and the specialty of the Arabic language itself (Lopez et al., 2011). Question Answering systems can be classified according to four dimensions based on the input question and answer type, data sources format, the scope of the domain being asked, and problems the system is trying to solve (Al Chalabi, Hani., 200).

This paper presents the build process of a question answering system that answers mainly Arabic factual questions that are closed to the prophet Muhamed pbuh biography domain with an educational perspective. As in figure 1, the system workflow begins by a student asking a question in his natural Arabic language. The system receives the question via a GUI and sends it to the question-answering module. This module is responsible for answering the question instantly for the learner, which he receives with a remark that it is an unvalidated answer by the educator. Then the QA module saves the question with the generated answer in a database. This question and answer are saved with a pending validation status. The instructor then receives these pending question/answer pairs to be validated. The instructor then has the ability to validate the answers generated by the system or correct them and send them back to the database, which saves them under a validated question/answer status. This pair is then sent to the learner that asked the question as your validated question/answer pair.

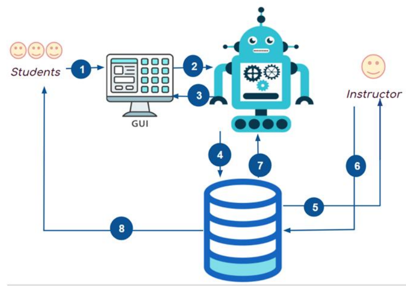

Figure 1. Edu. QAS Architecture

2. Methodology

The methodology presented here consists of 5 stages, as shown in Figure 2, data processing, question analysis, ontology mapping, query generation, and answer processing. The first stage is building the knowledge base underlying the system. The knowledge base is made up of Arabic ontology built on the Protégé tool as an RDF/OWL file. The second stage is classifying and analyzing the question generating question class, expected answer type, and question keywords. The third stage is mapping the question keywords with the underlying ontology elements. The fourth stage is generating the question pattern according to the mapped keywords with the expected answer type then generating the query templates according to the question templates. The fifth stage is ranking the answer received and presenting the selected answer. The following sections explain each step in detail.

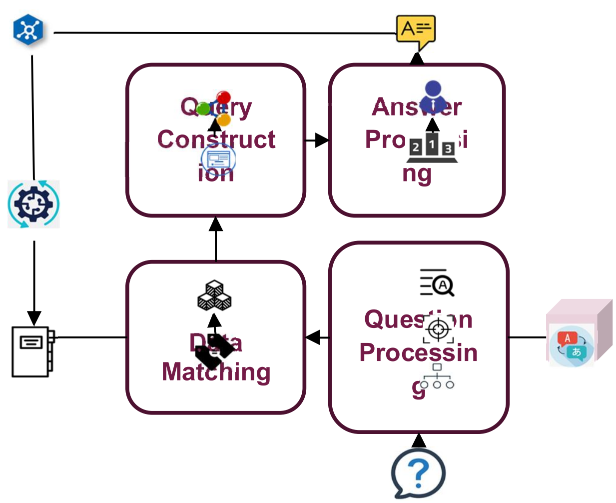

Figure 2. The question answering unit stages

3. Ontology

This ontology is designed with the help of an Arabic linguist to adjust its classes, properties hierarchy, and linguistic issues. It's written in OWL. The domain under study is represented as RDF triples (i.e.subject, predicate, object) for our QAS to answer factoid, yes/no, and definition questions in this domain. This ontology is not restricted to our system but can be used by other applications serving this special domain.

1. Isri Stemmer: http://oujda-nlp-team.net/en/programms/stemmer/
2. Alkhalil: http://oujda-nlp-team.net/en/programms/stemmer/
3. Farasa: https://farasa.qcri.org/
4. PY4J: https://www.py4j.org/

That's because the design of the ontology is made to be easily edited by adding or deleting Sera facts without defecting the ontology structure.

The ontology domain covers a small part of the prophet Muhammad pbuh Biography "Sera". It presents his family members' names, some of his friends, where and when he was born and died, his message and for whom this message, his "ghazawat", and the mosques he prayed in. It also presents info about the previous prophets and messengers from an Islamic perspective.

3.1. Building the ontology

Strict and clear steps are followed to define, design, and build our ontology from scratch. It started by determining the domain and the scope of the ontology. This ontology is built to serve under a question answering system. The questions that are expected to be asked are factoid as well as yes/no questions. The following questions are examples:

- هل تزوج النبي من أم سلمة ؟
- إلى أين هاجر محمد ؟
- من هو الصحابي الذي كان زوج فاطمة بنت محمد ؟

After defining the domain and the scope of the ontology, comes the design steps which are designing classes and subclasses. After that, we defined the relations of the object properties between these classes. Then triples are created by creating instances for these classes and asserting the object properties.

3.2. Ontology classes

Figure 3 presents the whole ontology classes. Most of the classes in the ontology are made to serve mainly in the answer ranking process where we can detect the type of the answer from the ontology and compare it with the answer type expected in the question classification process. For example, a question like: من هم قوم محمد ؟ "who are Muhammad people?". The expected answer should be of type "قوم " (a group of people or simply people) according to the question focus (I.e. first noun in the question as shown later).

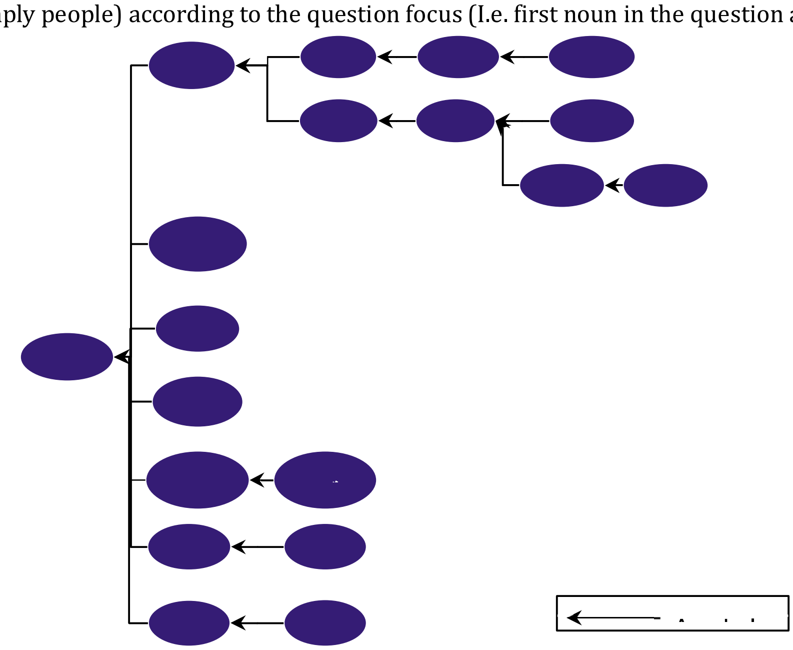

Figure 3. QAS classes tree

3.3. Classes properties

The properties serve as the predicates in RDF triples. A predicate connects between named individuals (instances) of the ontology. Figure 4 presents the ontology properties.

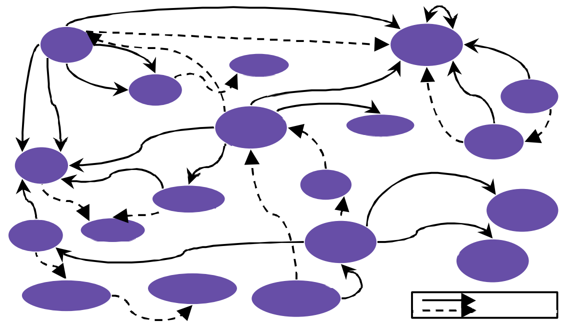

Figure 4. QAS object properties

3.4. Instance Creation

We created 95 instances that are classified according to classes types. Figure 5 depicts an example for the man type classification. Some of them are prophets, some are prophets and messengers, and some are companions.

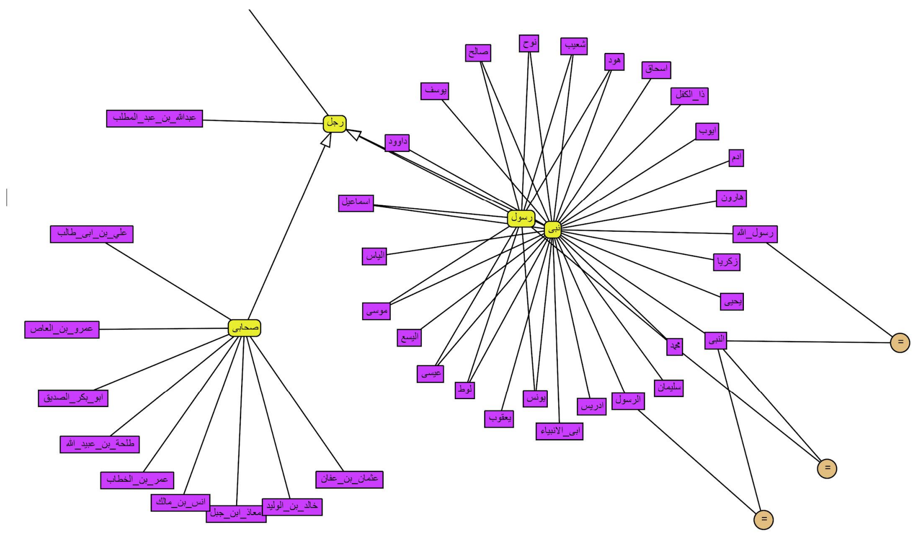

Figure 5. Example for the man type classification entities

4. Question Processing

The question processing module consists of a lexical analyzer module that morphologically processes the question to detect the question word, expected answer type, and question keywords. It begins by tokenization where the question is tokenized from being a whole string to an array of strings. So a question like "من هو الصحابي الذي كان زوج فاطمة بنت محمد ؟" is tokenized to [‘من’، ‘هو’، ‘الصحابي’ ،الذي، كان،زوج ، فاطمة ، بنت ، محمد ’؟’]. After that, the system detects the question focus. This question focus can help in expecting the answer type as shown later in the rule-based question classification section. The question focus in most cases is detected as the first noun or a noun phrase in the question. For example, in the question " ما هي المساجد التي صلى فيها النبي ؟" (What are the mosques where the Prophet prayed?), the question-focus will be "المساجد" which is useful in this case. But in some cases where the first noun could be a general word like the word "اسم/تعريف" like "ما اسم المساجد التي صلى فيها النبي؟", the system rejects this word and takes the next noun word to be the question focus.

1. Isri Stemmer: http://oujda-nlp-team.net/en/programms/stemmer/
2. Alkhalil: http://oujda-nlp-team.net/en/programms/stemmer/
3. Farasa: https://farasa.qcri.org/
4. PY4J: https://www.py4j.org/

For Arabic language, factoid question classification takes place depending on the question words or the interrogative particles (IP) أدوات الأستفهام in Arabic. They are kam - كم , man -من , ma - ما , ayn -أين , mata -متى , ay -أي , kaif -كيف . To process the question, we have used a rule-based approach that detects the IP with some pattern attached to it in some cases. The first pattern is straightforward and doesn't need more processing except for detecting the question word "Mata", "Ayna", "Kam", and "Hal". The expected answer type to these questions is always time or date, place, number, yes or no respectively, which are classes in the ontology in their Arabic names (وقت، مكان، رقم) for the first three question words. The second pattern, detecting the answer type depends on both the IP and the question focus, as shown in Table 1 for the 2 IPs "kam" and "ay". Table 1 presents the first and second patterns with examples.

The third pattern is the pattern that asks for a definition of somebody or something. It starts with the IP "Mann" which is who in English, or "Ma" which correspondents to what in English, then a pronoun, then the person or thing needed to be defined. The fourth pattern is the same as the third adding other keywords to it. In this case, the answer type is detected according to the noun or the question focus that follows directly the pronoun (and or the relative pronoun) i.e. "هو / هي " or "الذي/التي" in the question. This pattern comes with the 2 IPs "ma" and "mann". Table 2 presents the third and fourth patterns with examples.

The fifth and the last ones come only with the IP "ma" where the expected answer could be determined according to the Question focus or the name/number of the question focus. In this last case, the question focus is detected as the first noun that comes after the word name or number. Table 3 presents the fifth pattern with examples.

Qword	متى	أين	هل	كم	أي
Patterns	[QW] + [KWs]				
Example	متى ولد النبي ؟	أين ولد النبي ؟	هل تزوج النبي من أم سلمة ؟	كم عدد بنات النبي ؟	
Pattern				[QW] + [QF] + [KWs]	
Example				كم امرأة تزوجها النبي ؟	أي كتاب مقدس جاء به الإسلام ؟
Pattern					PP + [QW] + [QF] + [KWs]
Example					الى أى مدينة هاجر محمد ؟
AType	وقت	مكان	Yes/NO	# of type QF	QF

Table 1. The first & second patterns in the question and answer type classification method

Ques word	ما	من
Patterns	[QW] + [HWA/HEYA] + [QF]	
	[QW] + [HWA] + [تعريف] + [QF] [QW] + [تعريف] + [QF]	
Example	ما هو الإسلام ؟	من هو عمر بن الخطاب ؟
Ans Type	Def of QF	

Table 2. The third and fourth patterns in the question and answer type classification method

Qword	ما	من
Patterns	[QW] + [عدد/اسم/اسماء] + [QF] + [KWs] [QW] + [HWA/HYA] +[اسماء/عدد] + [QF] +[KWs] [QW] + [عدد/اسم/اسماء] + [KWs] [QW] + [HWA/HYA] +[اسماء/عدد] +[KWs]	[QW] + [HWA/HEYA] + [KWs] [QW] + [HWA/HEYA] + [QF] + [KWs]
Examples	ما اسم الصحابي الذي تزوج من بنت النبي فاطمة ؟ ما عدد الغزوات التي غزاها النبي ؟ ما اسماء بنات محمد ؟ ما عدد زوجات النبي ؟	من هو ابن مارية القبطية ؟ من هو الصحابي الذي تزوج فاطمة ؟
AType	Person name or Number	Person (or group of people) QF (person)

Table 3. The fifth pattern in the question and answer type classification method

5. Ontology Mapping

The step of ontology matching is important to grasp the ontology terms that describe the question keywords. These items are considered to be used then to build the SPARQL query. Figure 2 shows our approach for ontology mapping. The input to the ontology mapper is a set of question keywords resulting from the question processing phase. The output is the question mapped elements with the question form that is expected to match one of our templates.

To do the process of ontology mapping in a proper way, two challenges were tackled. The first one is the difference in Arabic writing styles and text formats in which users can use different text formats with the same root for the same meaning. For example, "ارسل به" and "ارسله". The second is phrases rather that many of the ontology elements are made of more than one word. For example, the named individual "أبو بكر الصديق" is made up of 3 words rather than one word.

An n-grams method taken from [3] and adjusted to analyze the question keywords taking into consideration the mentioned challenges. A maximum number of n-grams is set to be 4, because no entity in the designed ontology is greater than a 4-word length. The algorithm begins with the highest n-gram, which is 4 words, then n is decreased until a matched entity is found. It began with the highest n-gram because it could be more specific. (Master paper ref) If there are still question keywords, a word stemmer is used and research the ontology for each stemmed word. The algorithm is as follows:

As an example, for the question "من هم بنات خديجة بنت خويلد؟",

- "خديجة بنت خويلد" matches an instance of class "ام المؤمنين"
- "بنات" matches an object property "بنت" after being stemmed.

5.1. Ontological Dictionaries Builder

Two dictionaries are created to help in doing the ontology mapping process in a faster manner. This dictionary contains elements of the ontology that are of types named entity, object property, or class mapped to their types like the example mentioned in Table4. That is, we do it without the need to query the database many times. The dictionaries are created to help in 2 tasks. First, to quickly search the ontology elements if any of the question keywords are matching some ontology elements. Secondly, if found a matching element, we can get its type from this dictionary without having to query the ontology. Table 5 shows an example of the resulting processed dictionary.

Ontology Element	Type
محمد بن عبد الله	NamedIndividual
ارسل_الى	ObjectProperty

Table 4. An example for the Types Dictionary.

Processed Ontology Element	Original Ontology Element
ارسل	ارسل_الى
ارسل	ارسل_ب

Table 5. An example for the Processed Dictionary.

6. Template based technique for querying

After finishing the ontology mapping phase, we get mapped entities that could be used for querying and getting the answer. To build the query with these mapped entities, the system needs to understand in which context they are mentioned and what the user is actually asking for. For this purpose, an analysis is done for a big number of different Arabic questions in the ontology context to get the different frequent patterns of the entity types for most of the wh-questions. This resulted in creating 6 question templates as in the following 3 tables (Tables 6, 7, and 8).

1. Isri Stemmer: http://oujda-nlp-team.net/en/programms/stemmer/
2. Alkhalil: http://oujda-nlp-team.net/en/programms/stemmer/
3. Farasa: https://farasa.qcri.org/
4. PY4J: https://www.py4j.org/

Question Patterns	"Object Property", "Named Entity"	"Class","Object Property", "Named Entity"	"Object Property", "Class","Named Entity"

Sparql Template	SELECT DISTINCT ?a
	WHERE {
	NE OP ?a
	UNION
	?a OP NE
	}		

Examples	اين ولد محمد ؟ Named Entity محمد Object Property ولد_في	أي كتاب مقدس انزل على محمد ؟ Named Entity Object Property Class	اين سكنت قوم ثمود ؟ Named Entity Class Object Property

Table 6. Question Patterns for the first query template.

Question Patterns	"Named Entity", "Object Property", "Named Entity"	"Class", "Named Entity"	"Object Property", "Object Property", "Named Entity"

Sparql Template	ASK{
	NE1 OP NE2
	}
		SELECT DISTINCT ?a
		WHERE {
		?a rdf:type CLSS
		{NE ?x ?a}
		UNION
		{?a ?x NE} }
			Apply Template 1 on {OP2,NE1}
			Apply Template 1 on {OP1,NE2} where NE2 resulted from the first query with

Examples	هل عائشة زوجة محمد ؟ NE2 OP NE1	من الصحابي الذي تزوج فاطمة ؟ NE CLLS	من هم أزواج بنات محمد ؟ Named Individual Object Property Object property

Table 7. Question Patterns for the second query template.

Question Patterns	"Class","Object Property","Named Entity", "Object Property", "Named Entity"	"Object Property","Named Entity", "Object Property", "Named Entity"
Sparql Template	If Template 2 {NE1, OP2,NE2} = TRUE: Apply Template 1 {OP1,NE1}	
Examples	من هو الصحابي الذي كان زوج فاطمة بنت محمد ؟ Named Individual Object Property Named Individual Object Property Class	من هو زوج فاطمة بنت محمد ؟ Named Individual Object Property Named Individual Object Property

Table 8. Question Patterns for the third query template.

By getting the pattern for the question mapped entities, the system is designed with a query template for each question pattern. According to the detected answer type in the question processing stage, we format the answer. The type of answer is detected using a function that queries the ontology for an array of its direct class and this class's parents. For example, if we have a question like "من هو زوج فاطمة ؟" which answer is "علي بن ابي طالب". Here in this example, the supposed answer type detected from the question analysis phase is "رجل". But the class for the actual answer we get is "صحابي" which is a class under class "رجل". So although the two types seem different, the answer is right and should not be rejected. So the best solution to this problem is to compare the assumed answer type from the question phase with each item in the array of the actual answer tree types rather than its direct parent type.

7. Evaluation

7.1. Results and discussion

A preliminary evaluation of our sera system is shown here, in which we prove that the system behaves as expected. The system was designed mainly to convert an Arabic language question to a SPARQL query using a set of predefined question patterns and their matching query templates as well. The system succeeded in answering 127 questions out of 160 questions. Table 9 presents the results with the phases the system passed by.

	Exp 1	Exp 2	Exp 3	Exp 4
Number of Questions	127	127	127	127
Correct Answers	48	81	92	106
Blank Answers	74	41	30	17
Incorrect Answers	5	5	5	4
Precision = # of correct ans / (# of correct ans + # of incorrect ans)	90.5%	94%	94.8%	96.3%
Recall = # of correct answers / # of questions	38%	63%	72%	83%
F-measure = 2*(P*R)/(P+R)	53%	75%	82%	89%

Table 9. System phases results.

7.2. Other contributions

A Rule-based question word extraction method is designed to solve the problem of POS tagging problems. Also, for better-stemming results, we tried (Python ISRI[1] stemmer, ALKhalil[2] java stemmer, and FARASA[3] stemmer) and found that Farasa stemmer was the best. But using Farasa, which is a java application from a python code where somehow a programming challenge to call a Java application from python. We used the py4j[4] tool for this task and wrote a tutorial about it.

8. Conclusion

This system can be adapted for any Arabic biographical ontology. The system also can be adjusted for Latin languages by adjusting the question classification method, question patterns, and changing the ontology, and the lexical language tools. The ontology mapping process could have been enhanced if we added question expansion techniques for the semantically different or derivative question keywords to match the ontology entities. For example, the question من هم أولاد النبى is not answered because it should be "أبناء" not "اولاد"Other question patterns could be observed and added by language experts to cover more question types and patterns. In the stop words removal process we had to add and delete some words to match our target. For example, we had to add هن ،هم، هل، متى) (اي and remove "بن''. when we tried to identify the question words using the pos tagger, we found that some question words are wrongly tagged such as من and آي. Comment clauses have to be considered in the question analysis phase because it can be mistaken in the ontology mapping phase. For example, صلى الله عليه وسلم is mistaken by the object property صلى في.

8.1. Theoretical and practical implications

We proposed a design for an educational online system that is able to use a QA system effectively. This system can help students foster their knowledge and test their understandings. This system can be adapted to be used by other Latin languages and for other biographical topics. So it's not restricted to this ontology topic nor the Arabic language. One would just follow the design steps and change the ontology, linguistic tools, and question patterns with their corresponding query templates. Also, the question classification method needs to be changed for other languages. Parts that need to be changed or adopted are highlighted by blue ovals in figure 6. It's also worth to be mentioned that the ontology can be adapted also and used for any other Arabic biographical domain. It's made from scratch following strict rules of building an ontology and with the help of Arabic linguistic experts.

A rule-based question classification method is adjusted from (Al Chalabi, Hani) which is able to better classify factoid Arabic questions as well as definition questions. A template-based answer processing method is created by using the ontology mapping method from (Alagha et al., 2015) and a template-based the query generation method that doesn't need semantic rules like (Alagha et al., 2015) did.

1. Isri Stemmer: http://oujda-nlp-team.net/en/programms/stemmer/
2. Alkhalil: http://oujda-nlp-team.net/en/programms/stemmer/
3. Farasa: https://farasa.qcri.org/
4. PY4J: https://www.py4j.org/

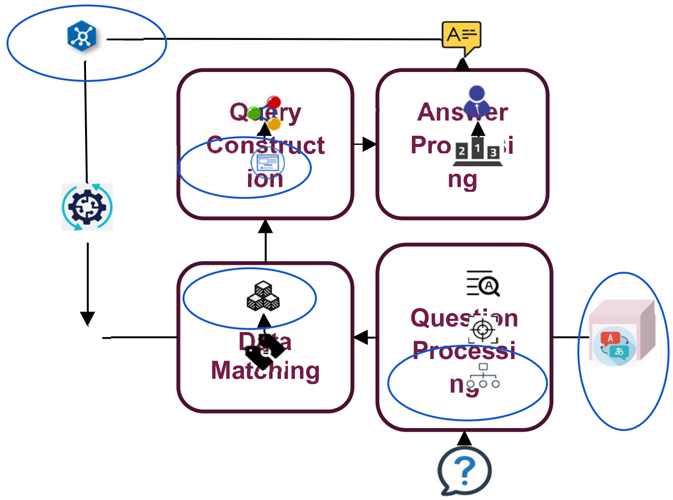

Figure 6. Parts of the system that needs to be changed when adopting the system

References

AbuTaha, A.W. and Alagha, I.M., 2015. *An Ontology-Based Arabic Question Answering System*. Islamic University-Gaza.

Al Chalabi, Hani., 2015. *Question Processing for Arabic Question Answering System*

Betts, George Herbert, 1910. *The Recitation*. Mount Vernon, Ia., The Hawk-eye publishing co.

Calijorne Soares, M. and Parreiras, F., 2020. *A literature review on question answering techniques, paradigms and systems.* Journal of King Saud University - Computer and Information Sciences, 32(6), pp.635-646.

Chin, C. and Osborne, J., 2008. *Students' questions: a potential resource for teaching and learning science. Studies in Science Education*, 44(1), pp.1-39.

Lampert Andrew, 2004. *A Quick Introduction to Question Answering*

Lopez, V., Uren, V., Sabou, M. and Motta, E., 2011. *Is Question Answering fit for the Semantic Web?: A survey. Semantic Web*, 2(2), pp.125-155.

Samy, H., E., E. and Shaalan, K., 2019. *Arabic Question Answering: A Study on Challenges, Systems, and Techniques. International Journal of Computer Applications*, 181(44), pp.6-14.