IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON ADVANCED DATA MINING METHODS FOR SOCIAL COMPUTING

Received October 23, 2019, accepted November 15, 2019, date of publication November 27, 2019,
date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2956233

Leveraging Linked Open Data to Automatically Answer Arabic Questions

MOHAMMAD AL-SMADI[1], ISLAM AL-DALABIH[1], YASER JARARWEH[1,2], AND PATRICK JUOLA[2]

[1]Computer Science Department, Jordan University of Science and Technology, Irbid 3030, Jordan
[2]Mathematics and Computer Science Department, Duquesne University, Pittsburgh, PA 15282, USA

Corresponding author: Mohammad Al-Smadi (masmadi@just.edu.jo)

ABSTRACT The interchangeably connected Web technologies and the advancements that accompany the semantic web content's leaps, have raised many challenges in the results' retrieval process especially for the Arabic Language. This research targets an important, yet insufficiently precedent, area in using Linked Open Data (LOD) for Automatic Question Answering systems in the Arabic Language. The significance of work presented, comes from its ability to overcome many challenges in querying Arabic content. Some of these challenges are: (a) bridging the gap between natural language and linked data by mapping users' queries to a standard semantic web query language such as SPARQL, (b) facilitating multilingual access to semantic data, and (c) maintaining the quality of data. Another challenging aspect was the lack of related work and publicly available resources for Arabic Question Answering Systems over Linked Data, despite the vastly growing Arabic corpus on the web. This paper presents a novel approach that targets Automatic Arabic Questions' Answering Systems whilst bypassing many featured challenges in the field. A hybrid approach that evaluates the effectiveness of using LOD to automatically answer Arabic questions is developed. The approach is developed to map users' questions in Modern Standard Arabic, to a standard query language for LOD (i.e. SPARQL) through: (i) extracting entities from questions and linking them over the web using Named-Entity Recognition and Disambiguation (NER/NED), and (ii) extracting properties among extracted named entities using a dependency parsing approach integrated with Wikidata ontology. To evaluate our proposed system, an Arabic questions dataset was created including: (a) Question body in Arabic language, (b) Question type, (c) SPARQL Query formulation, and (d) Question answer. Evaluation results are promising with a Precision of 84%, a Recall of 81.3%, and an F-Measure of 82.8%.

INDEX TERMS Semantic web, question answering systems, structured data, natural language processing, Arabic language.

I. INTRODUCTION

The web has become the most important resource for digital knowledge. Recently, there is a trend to restructuring the web into representing data rather than representing documents. Therefore, a huge amount of semantic data is now available on the web using semantic web meta-data (i.e. in Resource Description Framework (RDF) and Web Ontology Language (OWL) formats). The Semantic Web as the new vision of the web 3.0, builds on the idea of enriching the web-links with meaningful properties describing linked documents' relationships [1]. Moreover, Semantic Web is based on structured data and well-defined relationships representing meanings of data.The main goal of the semantic web is to help machines understand what is represented. In the case of Question Answering Systems (QASs), structured data needs to be explored to return precise and short answers.

The Linked Open Data (LOD) [2], [3] is a project that started in 2007, with a goal to have interlinked, open and structured data of different domains as billions of web entities are being published and interlinked rapidly. According to the LOD cloud web page,[1] the LOD cloud consists of 1,239 datasets with 16,147 links (as of March 2019).

OWL and RDF are description logic languages that are used for representing relationships among data on the Web

The associate editor coordinating the review of this manuscript and approving it for publication was Yongqiang Zhao.

[1]https://lod-cloud.net/#about

using ontologies. OWL and RDF make the web content readable for machines and useful for many applications such as automated questions' answering systems [4], [5]. RDF Schema (RDFS) is a language that can be used to define a vocabulary for describing classes, sub-classes and properties of RDF resources [6]. SPARQL Protocol and RDF Query Language (SPARQL), is a query language used to retrieve data out of semantic web ontologies.

Being a part of LOD; structured knowledge bases such as Freebase and DBpedia have become more popular and are used in many applications. One of these applications is Question Answering. Question Answering Systems (QASs) have received wide attention recently; as conventional search engines are not capable of providing exact and precise answers for questions [7]. QASs give precise and accurate results compared to the information retrieval applications such as web search engines. The web search engines allow the users to submit a query, and thereafter return a candidate answer as a document that contains information relative to that specific query, not the answer itself. Web search engines such as Google, Yahoo, and MSN [8] use a keyword matching technique between the user's query and the web documents, which may lead to many incorrect or undesired answers [9].

LOD employs RDF to structure the data over the web [10]. RDF annotates the data as triple syntax (subject-predicate-object). It annotates resources in a way that enables the machine to understand the information. RDF uses Uniform Resource Identifiers (URIs) to represent resources [11]. RDF Query language (SPARQL) is used to query the data and retrieve answers from RDF [6]. QAS over LOD is not a trivial task due to the challenge of mapping the input questions into RDF triples' format. The formed triples are then used to build a SPARQL query and get the correct answer.

Building a QAS requires several services including Natural Language Processing (NLP), Information Retrieval (IR), Database Administration (DBA), and Artificial Intelligence (AI) [12]. QASs are categorized into two types: open domain and closed domain. Open domain QASs works on the general question and common knowledge, while the closed domain QASs answer questions based on specific domain knowledge [13]. Most of the QASs focus on unstructured/semi-structured textual data resources to extract answers using information retrieval techniques. Recently, and with the growing amount of semantic knowledge represented in LOD cloud; a remarkable increase of interest for building LOD-based QAS has taken place in the research field. However, LOD-based QAS requires a mapping service to map natural language questions into a machine-understandable queries (i.e. structured query language such as SPARQL) [14].

QAS over linked data has many challenges. These challenges include: (a) bridging the gap between natural language (i.e. Arabic) and linked data by mapping the user query to a standard semantic web query such as SPARQL, (b) facilitating multilingual access to semantic data, and (c) detecting the retrieved data quality [15].

With the huge attention drawn towards LOD recently, there exists the need for benchmark datasets for question answering systems built over LOD. Therefore, the ''Question Answering over Linked Data'' (QALD) benchmark dataset was established in 2011.[2] QALD is a benchmark dataset that aims to evaluate question answering systems developed over LOD. Moreover, it helps in developing useful systems and methods that can deal with the huge amount of available RDF data and automatically answer users' questions over the web [16].

Most of previous research on QAS over LOD (QALD) focused on the English language. To the best of our knowledge, there is few research available for the Arabic language. One of the most important challenges over LOD is the lack of related work and publicly available resources. This is due the special traits of the Arabic language. The Arabic language is a complex and morphologically rich language [17]. It has many challenges such as no capitalization, agglutination, optional short vowels, free word order, lack of uniformity in writing style, and lack of linguistic resources.

This study introduces several contributions, including (a) Studying the effectiveness of using the LOD in Arabic QAS, (b) Extracting answers for Arabic questions by transforming them to SPARQL Query language, and (c) Building a dataset of Arabic pairs of Questions and Answers with their corresponding SPARQL query. This dataset can be used as a benchmark dataset for Arabic-language QAS over LOD which may help researchers working on the problem to evaluate and compare their work.

The rest of this paper is organized as follows: Section II sheds the light on related work for QAS and QAS for Arabic language, Section III explains the dataset collection and preparation, Section IV explains the research method and the proposed approach, Section V presents the approach's results and findings. Section VI discusses the main findings of the evaluation results. Finally, Section VII concludes this work and presents future work plans.

II. LITERATURE REVIEW

This section introduces literature review over QASs and the benchmark that evaluates the QASs. In addition, it presents a comparative study of existing QAS for Arabic language and other languages. This section is divided into two main parts: (a) Question Answering Systems over LOD, and (b) Existing Question Answering Systems for the Arabic Language.

A. QUESTION ANSWERING SYSTEMS OVER LINKED OPEN DATA

In the context of Question Answering, a lot of systems that are based on LOD have been proposed in the past years. Most of them were developed for the Open Challenge on Question Answering over Linked Data (QALD[3]) which has different versions from QALD-1 to QALD-7.

[2]http://qald.aksw.org/
[3]https://qald.sebastianwalter.org/

In the first version of QALD (QALD-1), two resources of data are used: DBpedia [4] and an RDF export of the MusicBrainz [5] Database. 100 questions (50 for training and 50 for testing) were prepared with corresponding SPARQL queries. The second year of the challenge produced the second version, (QALD-2). The training set and the testing set of the first challenge were combined to build a new training set (100 questions) and 100 new questions were created for testing. The following mentioned systems are examples of the systems that participated in QALD-1 and QALD-2 respectively. QALD-1 systems are: FREyA [18], PowerAqua [19], TBSL [20], and Treo [21]. QALD-2 systems: SemSeK [22], BELA [23], and QAKiS [24].

FREyA [18] is the first system developed that participated in the QALD challenge. It is an interactive natural language Interface for querying ontologies. The approach was used for interpreting the input question composed of the syntactic parsing and ontology-based lookup. The evaluation of FREyA system was reported using the MusicBrainz and DBpedia datasets, and achieved an f-measure of 0.58 and 0.71 with respect to the datasets.

The SemSeK system [22] focuses on how to interpret the natural language queries to generate SPARQL queries. The interpretation processes that are used include a deep linguistic analysis, semantic similarity/relatedness and query annotation for the concepts of LOD (classes and instances). SemSeK evaluates their approach on a QALD-2 dataset and obtained f-measure of 0.46, a precision of 0.44, and a recall of 0.48.

Another system that participated in QALD-3 is the SWIP [25] system. The SWIP translates the natural language query into a SPARQL query using two main steps: interpreting the natural language query into a pivot query, then translating the pivot query into a formal query. In the first step, the system identifies the query, then extracts the relation between sub-strings of the natural language query using dependency parsing. After that, the pivot query is generated and mapped into predefined query pattern. Finally, the generated query is ranked and reformulated to be proposed to the user. The system was evaluated using QALD-3 dataset and achieved a precision of 0.16, recall of 0.15, f-measure of 0.16 on DBpedia.

The Xser system [26] returns the answer for the input natural language question in two phases. Firstly, the predicate-argument-structure is detected using the semantic parser. Next, the query is matched into a knowledge base using the Directed Acyclic Graph (DAG) dependency parsing. The Xser approach is evaluated on the QALD-4 test dataset and is obtained an f-measure of 0.72, a precision of 0.72 and a recall of 0.71 over 50 questions. gAnswer [27] also evaluated their system on the QALD-4 dataset. The system used a graph-based approach that answers the question in two steps. In the first step, is transforming the natural language query into web semantic based graphs using dependency parsing. The second step matches the generated graph into RDF dataset. The gAnswer system achieved an f-measure of 0.40, a precision of 0.40, and a recall of 0.40.

The QAnswer system [28] retrieves answers of the user questions from the DBpedia dataset. It used a Wikipedia-based approach to extract lexicalizations of the DBpedia entities that are matched with the input question. The QAnswer system tested on the QALD-5 dataset and achieved an f-measure of 0.30.

Pouran-ebn veyseh, in [29] , proposed a cross-lingual QAS using a unified semantic space among languages. The proposed approach consists of different steps: keyword extraction, entity linking, type extraction, and relation selection. The system was evaluated on DBpedia dataset with English, Persian and Spanish languages using 49 questions for Persian and Spanish languages. The reported results obtained an f-measure of 0.65 a precision of 0.55 and a recall of 0.53 for the English language. The Persian obtained a precision of 0.53, recall of 0.51, and f-measure of 0.52, while the Spanish Language obtained an f-measure of 0.54.

The GFMed [30] proposed a domain-specific QAS for Biomedical interlinked data. GFMed used a grammatical framework based on Description Logic to build controlled natural language interface targeted towards biomedical information with the ability of mapping queries into their corresponding SPARQL format.

In 2017, the systems AMAL, and ''Sorokin and Gurevych'' were developed for QALD-7 [31]. AMAL is a QAS for the French language. The proposed approach consists of different stages. In the first stage, the pattern matching was used for question classifying into types (e.g. Boolean or Entity). After that, the entities and properties were extracted based on syntactic parsing and the entities were linked to DBpedia. In addition, the Wikipage disambiguation links were used to help. The identification process of properties was done using a manually created lexicon which contains the common DBpedia properties. The system was evaluated and achieved a precision of 0.720, a recall of 0.720, and f-measure of 0.720.

Sorokin and Gurevych [32] converted a natural language question into SPARQL by generating candidate semantic representation for a question and comparing them using a Convolutional Neural Network (CNN). The system was evaluated on Wikidata dataset and obtained a precision of 0.3507, a recall of 0.4318, and f-measure of 0.3640.

WDAqua-core1 [33] is a multilingual QAS for RDF Knowledge Bases. The proposed approach takes a question and returns it in SPARQL language format in four steps: Query Expansion, Query Construction, Query Ranking, and Answer Decision. It depends on the semantics of the question instead of the syntax. The system was evaluated on QALD-7 over five languages, including English, French, German, Italian and Spanish, and obtained a precision of 0.63, a recall of 0.32, and f-measure of 0.42 for the English language.

[4]https://wiki.dbpedia.org/
[5]https://musicbrainz.org/

B. QUESTION ANSWERING SYSTEM FOR ARABIC LANGUAGE

In this section, we review the research based on different techniques used for Arabic question answering systems.

Mohamed et al, 1993 developed the first Arabic answering system called Arabic Question-Answering System (AQAS) [34]. Their work is a closed-domain system which focuses on radiation domain. Furthermore, their work is considered a knowledge-based system that returns answers only from structured data (frames). However, there were no results published in the paper.

During the years from 1993 until 2002,the research on Arabic QASs witnessed a recession period with no related research publications. In 2002, Hammo et al. proposed QARAB [35]. Their proposed system uses techniques from Information Retrieval (IR) and Natural Language Processing (NLP). QARAB takes input from the user expressed in Arabic language and tries to retrieve short answers. This system uses unstructured documents which were collected from Al-Raya newspaper in Qatar to extract answers. The system did not address all types of questions such as How (كيف/ kyf) and Why (لماذا/lmA*A), but it handles the input question as a ''bag of words'' that is searched for the index file to get a list of ranked documents that may contain the answer. The authors did not report any results or evaluation criteria (i.e., recall, precision). The evaluation was done directly by the four native Arabic speakers who asked the system 113 questions and judged the correctness of the answers for themselves.

Another approach was proposed by Benajiba et al., [36] which presented an Arabic QAS that works under the Java Information Retrieval System (JIRS) and a passage retrieval module called ArabiQA. This system used Arabic Named Entities Recognition (NER) system called ANERsys to identify proper names in the question. It focused on answering the factoid questions. The test set consists of 200 questions and 11,000 documents from Arabic Wikipedia. This system was evaluated and achieved a precision of 0.833.

Focusing on the systems that answer a particular category of questions, there is a QASAL system [37] proposed by Brini et al.,trying to answer the factoid questions (i.e., Who (من/mn), When (متى/mtY)). Nooj platform was used to identify the answers from a set of educational books. QASAL is composed of three modules: Question analysis, Passage retrieval, and Answer extraction. Bekhti et al. proposed an Arabic Question-Answering System (AQuASys) [38] to answer the factoid questions according to any expected type of answers: person, location, organization, etc. The system returned the most accurate answer and a list of ranked candidate answers. The evaluation process used 80 questions for testing and obtained an overall recall rate of 0.975 and a precision rate of 0.6625.

DefArabicQA (Arabic Definition Question Answering System), proposed by Trigui et al. [39] in 2010, can extract accurate answers for Arabic definition questions. This system uses a lexical patterns approach to retrieve precise and accurate definitions of organizations using Web resources. It uses a set of rules to classify definitions as correct or incorrect. After extracting the definition, the system ranks the results according to specific criteria; including Pattern weight, Snippet position, and the sum of word frequencies in the extracted definition. The evaluation process uses Mean Reciprocal Rank (MRR), by assigning a score for each question based on the first string in the correct answer. The authors used 50 definition questions only for testing in two experiments. The MRR was 0.70 in the first experiment and 0.81 in the second one.

In 2017 AlQuAnS [40] presented a system which combines different algorithms used in QAS to create a novel approach to the process of answer extraction. It used the Explicit Semantic Approach (ESA) in the passage retrieval process for passages ranking. The system obtained an accuracy of 0.222.

In [41] the proposed QAS answers Arabic question in two phases: a question processing phase and a document retrieval phase. In the question processing phase, the system classifies the questions using a Support Vector Machine (SVM) classifier. After that, the classified Arabic question is translated into a query that can get the answer from Wikipedia using a combination method of Arabic Part-of-Speech (POS) tagging and Arabic WordNet.

In the context of using the semantic web in Arabic QAS. AbuTaha et al., [42] proposed a rule-based and closed-domain system which answers Arabic questions by translating them into a SPARQL query. The system was evaluated on a pathology ontology that was manually created. They used 30 questions for testing, of which the system answered 28 correctly.

In addition, [43] presented an approach to map a SPARQL query to an Arabic sentence. The mapping process combines both morpho-syntactic analysis and English dependencies to generate an understandable Arabic query. The proposed approach consists of several steps including: (a) Extract the Arabic translations of terms in the SPARQL query, (b) Restructure the terms into valid sentences using Stanford dependencies of the English language, and a set of handcrafted rules, (c) Use of morpho-syntactic analysis and NLP techniques to boost the linguistic realization of the sentence, and (d) improve the legibility of the sentence by eliminating aggregation and redundancy.

In [44] , the authors proposed an approach to map Arabic natural language queries into SPARQL format using linguistic analysis. The first step of the mapping process is Noun Phrases (NPs) extractions using language parser. After that, the relations between NPs are identified. In the next steps, the extracted NPs and their relations are matched against the underlying ontology (Diseases ontology). Finally, the RDF triple is generated.

Table 1 presents a comparison between some existing Arabic QASs. Systems are compared according to their: covered questions types, used dataset, the system type (open or close), and if the system is a web-based system or not. In contrast to the available QASs for Arabic questions; our system's

TABLE 1. Comparison between some existing Arabic QASs.

Arabic QAS	Question types	Dataset	Web-based	Open do-main
AQAS [34]	-	Answer questions from structured data stored into a Database	No	-
QARAB [35]	Factoid questions	113 Factoid questions and Documents collected from Al-Raya Newspaper	No	No
ArabiQA [36]	Factoid questions	11,000 documents of the Arabic Wikipedia and 200 questions and a set of correct answers	No	Yes
QASAL [37]	A Factoid and Definitional questions	Collection of Tunisian books as a corpus and 43 definition questions	No	Yes
AQuASys [38]	Factoid questions	80 questions of different types	No	Yes
DefArabicQA [39]	Definitional questions	Web as the data source and 50 Definition questions	No	Yes
Our system	Factoid, Definitional and Yes/No questions	DBpedia and Wikidata as a data source and 400 questions of different types	No	Yes

significance is apparent in: (i) covering different types of questions (i.e. Factoid, Defintional, and Yes/No Questions), (ii) being evaluated using the largest dataset available for Arabic QASs (i.e. 400 questions), and (iii) to the best of our knowledge we are the first to leverage LOD in automatically answering different types of Arabic questions presented in natural language.

III. DATASET PREPARATION AND COLLECTION

To the best of our knowledge, there is no Arabic benchmark dataset for evaluating Question Answering Systems developed using LOD. Therefore, a dataset was prepared for this research purposes. The dataset was prepared and collected manually. The dataset covers different domains (History, Disease, Geography, Music, Food and Recipes, Sport, and Art) and has different types of Arabic questions provided with their corresponding SPARQL query and answers.

The dataset consists of 400 questions divided into three categories:

1) Factoid questions: this category includes questions that start with interrogative particles such as: (أين/Ayn/Where, في اي/fe-Ay/In which, متى/mtY/When, كم/km/How many, من/mn/who, ما/mA/what. This category has a total of 228 questions in the dataset.
2) Definition questions: this category includes questions that start with interrogative particles such as: (ما/mA/what, من/mn/who). This category has a total of 162 questions in the dataset.
3) List questions: this category includes questions that start with verbs such as (اذكر/A*kr/List). This category has a total of 10 questions in the dataset.

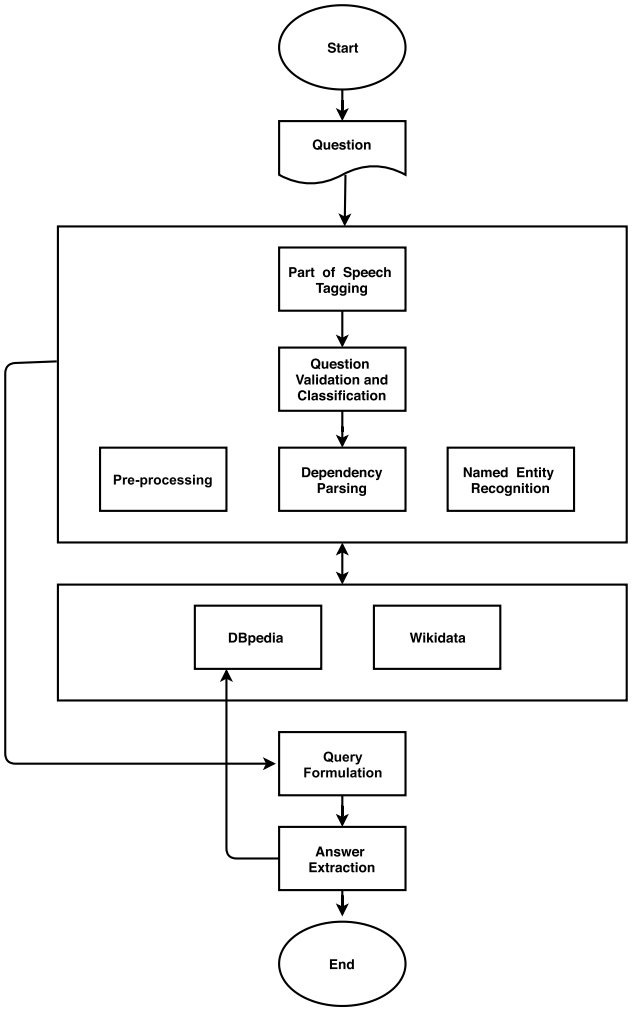

FIGURE 1. Overview of the research methodology phases.

Other questions' categories such as: Causal questions (ماذا/mA*A/What), Method questions (كيف/kyf/How), and Purpose questions (لماذا/lmA*A/Why) are not included in the prepared dataset. Table 2 summarizes question's categories, questions' types, their availability numbers, and provides an example from the dataset on each question's type.

IV. RESEARCH METHODOLOGY

In order to tackle this research problems and to leverage LOD in automatically answering Arabic questions, three main approaches were developed: (i) the baseline approach, (ii) a Wikidata-based approach, and (iii) a dependency parsing based approach. For each of the developed approaches, three main modules are implemented (see Figure 1):

1) Question Analysis: includes, question validation, resource identification, and property identification. Some NLP techniques are also used for analyzing the question, including Tokenization, Part of speech tagging (POS), Named-entity recognition (NER), and Dependency parsing.

TABLE 2. The categories for the dataset Arabic question based on question Interrogative Particles (IP).

Question Category	Question Type	Number of questions	Example
Factoid	اين/Ayn/Where	28	اين توفي ارسطو؟/Ayn twfy ArsTw?/Where did Aristotle die?
	في اي/fe-Ay/In which	8	في اي مدينة يوجد المسجد الاقصى؟/fy Ay mdynp ywjd Almsjd AlAqsY?/In which city is Al-Aqsa Mosque?
	متى/mtY/When	17	متى تأسست الاردن؟/mtY t>sst AlArdn? /When was Jordan established?
	من/mn/Who	38	من هو مؤسس مايكروسوفت؟/mn hw m&ss mAykrwswft? /Who is the founder of Microsoft?
	ما/mA/What	120	ما هي عاصمة الاردن؟/mA hy EASmp AlArdn?/What is the capital of Jordan?
	كم/km/How many	17	كم يبلغ عدد سكان السودان؟/km yblg Edd skAn AlswdAn? /How many inhabitants does Sudan have?
Definition	من/mn/Who	118	من هو بيكاسو؟/mn hw bykAsw? /Who is Picasso ?
	ما/mA/What	44	ما هو سيليو؟/mA hy sylyw?/What is Celio?
List	Non-IP	10	اذكر ابناء هيلاري كلينتون؟/A*kr AbnA' hylAry klyntwn./List the children of Hillary Clinton.

2) Query Formulation: implements a template-based method that utilizes linked-open data to map an Arabic question into a SPARQL query.
3) Answer Extraction: extracts answers from DBpedia using the generated SPARQL query.

The next sub-sections explain the three developed modules in more details.

A. QUESTION ANALYSIS

In the question analysis phase, the intent of the user query is interpreted, and then the question is mapped into a SPARQL Query. The output of this phase will be represented as an RDF triple. As depicted in Figure 2, this phase consists of four sub-processes: (i) question validation (ii) question pre-processing, (iii) resource identification, and (iv) property identification.

1) QUESTION VALIDATION

In this sub-process, we check the validity of the input question to ensure that the question format and the question type are valid. More precisely, the question must have a specific format, for instance: WH-questions must start with interrogative particle and end with the question mark, Yes/No questions start with the keyword (هل/hl/Do) and ends with a question mark, and the questions with no interrogative particles must start with a Verb (i.e., اذكر/A*kr/List). In order to validate

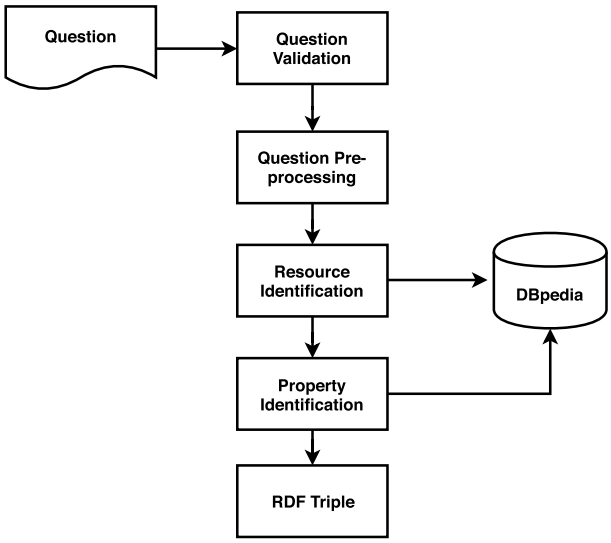

FIGURE 2. Question analysis sub-processes.

TABLE 3. POS tags that are used in the question validation process and their description.

Tag	Description
WRB	WH-adverb
WP	WH-pronoun
VB	Verb, base form
WRB	WH-adverb
VBP	Verb, non-3rd person singular present

questions, Part of speech tagging (POS) is used to extract the category (Verb, Noun, etc.) of the question tokens. Every word in the question is tagged with a corresponding POS tag (such as: nouns, verbs, adjectives, adverbs, pronouns, etc.). The POS tag is used to check if the first token in the input query involves a WH-pronoun such as when/متى/mtY, a verb such as اذكر/A*kr/List or WH-adverb such as how/كم/km. Table 3 presents the POS tags used to validate the question and their type. Queries are considered to be valid questions if they start with one of the POS tags presented in Table 4.

2) PRE-PROCESSING

In this step, a Java-based library called AraNLP [45] is used to tokenize, normalize, and stem the input questions. The pre-processing steps are explained in details as follows:

- **Tokenization and Normalization:** The tokenization process is used to segment a question into tokens and the normalization process is used to remove "HAMZA/ء" from the "ALEF/ا" (i.e. the "أ إ آ" are all replaced with the abstract letter "ا"). Arabic diacritics (Tashkeel) (such as "ً ٌ"), punctuation, and special symbols such as "\ $ # ? " are also removed.
- **Stemming:** The process of segmenting and separating affixes from a stem to produce prefix, stem and suffix parts for each word.
- **Named Entity Recognition:** Named Entity Recognition (NER) is an important process in the question

TABLE 4. The templates of SPARQL query that uses on the resource identification process.

Identification using equality operator	PREFIX rdfs:<http://www.w3.org/2000/01/ rdf-schema#> SELECT DISTINCT ?uri WHERE { ?uri rdfs:label ?var. FILTER (LANG (?var) ='ar'). FILTER (?var= 'Named Entity'@ar). }
Identification using regular expression	PREFIX rdfs:<http://www.w3.org/2000/01/ rdf-schema#> SELECT DISTINCT ?uri WHERE { ?uri rdf:type NER_Class. ?uri rdfs:label ?var. FILTER (LANG (?var) ='ar'). FILTER REGEX (?var, 'Named Entity'@ar).} }

analysis phase. The main goal of NER is to extract the proper nouns or entities from a question and classify them into specific categories such as Person, Location, Organization, Date and Time, etc. A Java library called FARASA [46] was used to extract named entities out of the Arabic queries. FARASA tags only the basic classes (Person, Location, and Organization).

3) RESOURCE IDENTIFICATION

Using our previous research [10], extracted entities are then linked to their corresponding DBpedia resources to get the resource URI that represents a subject or an object in the RDF triple. This task is done by sending a SPARQL query to the DBpedia endpoint [6] (see Table 4). The challenge of the lexical gap between the Arabic question keyword and the DBpedia vocabularies and resource names is resolved using the ''rdfs:label'' that retrieves the Arabic label for the DBpedia resources.

For example, when interpreting the question ''ما عاصمة الاردن؟/mA hy EASmp AlArdn?/What is the capital of Jordan?'', with respect to the DBpedia dataset, the extracted named entity ''الاردن/AlArdn/Jordan'' needs to be mapped into the resource ''http://dbpedia.-org/resource/Jordan''. Following the proposed approach in [10], the identification process consists of several steps as follows:

1) Identification using the equality operator: firstly, the named entities are extracted to be matched against the DBpedia resources using a SPARQL query that contains (FILTER (?var= 'Named Entity'@ar)). For example, to identify the named entity ''الاردن/ AlArdn/Jordan'' the SPARQL query shall contain (FILTER (?var= 'الاردن'@ar)) (see Figure 3):
2) Identification using regular expressions: in case the previous step returns null, we send another SPARQL query. This other query matches the extracted named entity against DBpedia resources label using a regular expression (FILTER REGEX (?var, 'Named Entity'@ar)) and restricts the resource to the named entity class using (rdf:type).
3) Text Similarity: in case the previous step (2) returns more than one resource label, we need to rank the candidate resources using text similarity and extract the most similar resource label to the named entity (see Figure 4). For instance, ''اين يقع نهر النيل؟/Ayn yqE nhr Alnyl?/Where is the Nile?''; the named entity ''النيل/Alnyl/Nile'' does not match any DBpedia resource. In this case a SPARQL query that contains (FILTER REGEX (?var, ' النيل '@ar)) is used to return all DBpedia resources that contain ''النيل''. Then, we calculate the string similarity score between the SPARQL query results and the named entity ''النيل/Alnyl/Nile'' and retrieve the most similar DBpedia resource (the highest similarity score). For the text similarity process, we used the Cosine Similarity algorithm that calculates the similarity between two strings by finding the cosine of the two strings after being transformed into vectors of occurrences. This task was done using a java library called java-string-similarity.[7]
4) If the result of all the previous steps is null, the system returns no answer for the input question.

```
REFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?uri
WHERE {
?uri rdfs:label ?var.
FILTER (LANG (?var) ='ar').
FILTER (?var= ' الأردن '@ar).
}
```

FIGURE 3. Example for the resource identification using the equality operator.

4) PROPERTY IDENTIFICATION

The property is also known as a predicate that is used to describe some aspects of the subject. In the RDF triple, the property is required to identify the relationship between the Subject and the Object. Figure 5 shows the properties ''capital'' and ''country'' that connects the resources (Jordan and Amman) in the RDF triple.

All properties connected with the identified resource (in-link and out-link) are obtained using a SPARQL query. For example, as shown in Figure 6 the resource of ''Jordan'' has ''البلد/Albld/country'' as an in-link property and ''عاصمة/EASmp/capital'' as an out-link property. The following SPARQL query is used for retrieving all properties for the specific resource.

SELECT ?property WHERE {{Resource ?property ?x.} UNION {?x ?property Resource}}

[6]http://dbpedia.org/sparql

[7]https://github.com/tdebatty/java-string-similarity

Question in Arabic: اين يقع نهر النيل؟
Buckwalter Transliteration: Ayn yqE nhr Alnyl?
Question in English: where is the Nile river located?
NER: اين/O, يقع/O, نهر/O, النيل/B-LOC
Named entity: النيل

SPARQL query:

```
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?uri
WHERE { ?uri rdf:type < http://dbpedia.org/ontology/Location>
?uri rdfs:label ?var.
FILTER ( LANG (?var) ='ar').
FILTER REGEX (?var,'النيل'@ar). }
```

Candidate resources: [منطقة أعالي النيل, نهر النيل, جسر النيل الأزرق, ولاية النيل الأزرق, ولاية نهر النيل, النيل الأبيض, ولاية النيل الأبيض, النيل الأزرق, الناصر (أعالي النيل), أعالي النيل (ولاية), كوبري قصر النيل]

The most similar DBpedia resource : نهر النيل
DBpedia resource: <http://dbpedia.org/resource/Nile>

FIGURE 4. Example for the resource identification using regular expression.

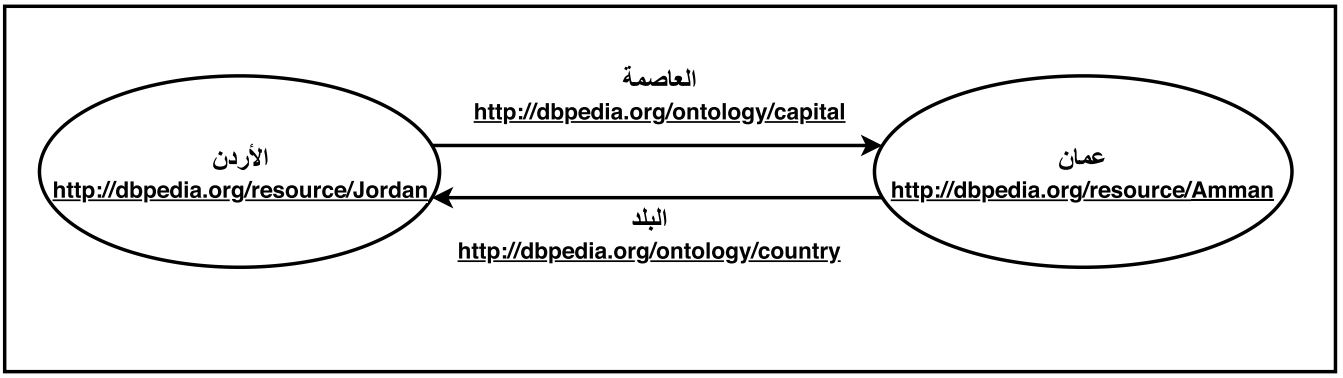

FIGURE 5. An example for in-direct and out-direct property for the Jordan resource.

Unfortunately, DBpedia does not have Arabic labels for most of its represented properties. Therefore, we propose three approaches to address the problem of property extraction: (a) baseline approach, (b) Wikidata-based approach, and (c) Dependency Parsing-based approach. The followed approaches depend on the extracted NERs which represent the RDF triple subject and/or object. The approaches are discussed in more details in the next section.

B. PROPOSED APPROACHES FOR PROPERTY IDENTIFICATION

In order to leverage LOD in answering Arabic questions, the user's questions are mapped into RDF triples including Subject, Property, and Object. For extracting the subject/object resources out of the user's questions, the conventional NER and entity linking techniques are used. Whereas, to extract the predicates/properties out of the input question and link them to the DBpedia corresponding predicates/properties, our research proposes three novel techniques as follows:

1) BASELINE APPROACH

In this approach, the RDF triple is built based on the expected Range and Domain of the property. The property range and domain are instances of ''rdf:Property''. The range is used to define the class or datatype for the Object of the RDF triple, while the domain is used to define the class or datatype

TABLE 5. Expected property range based on question type.

Question type	Range
من/mn/who	http://dbpedia.org/ontology/Person
{اين/Ayn,في-اي/fy-Ay}/where	http://dbpedia.org/ontology/Location http://dbpedia.org/ontology/Place http://dbpedia.org/ontology/City http://dbpedia.org/ontology/Country
متى/mtY/when	http://www.w3.org/2001/XMLSchema#date http://www.w3.org/2001/XMLSchema#gYear
كم/km/{How much, How many}	https://www.w3.org/2001/XMLSchema#double https://www.w3.org/2001/XMLSchema#integer http://dbpedia.org/datatype/squareKilometre http://dbpedia.org/datatype/inhabitantsPer-SquareKilometre http://www.w3.org/2001/XMLSchema#non-NegativeInteger
ما/mA/which	http://dbpedia.org/ontology/Location http://dbpedia.org/ontology/City http://dbpedia.org/ontology/PopulatedPlace http://dbpedia.org/ontology/Place http://dbpedia.org/ontology/Country http://dbpedia.org/ontology/Organization

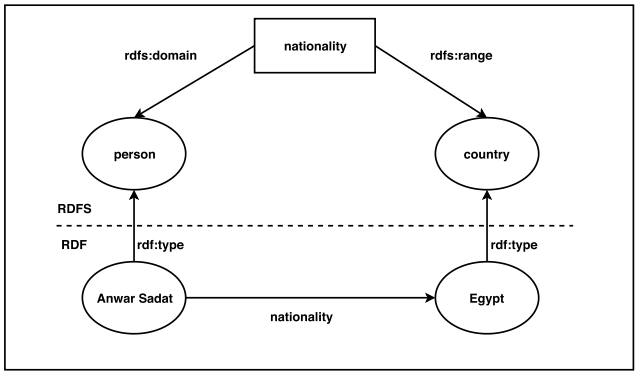

FIGURE 6. The domain and range for the "nationality" property.

for the Subject of the RDF triple. The expected range is extracted based on the question type (see Table 5). Whereas, the expected domain is extracted using the "rdf:type" of the extracted resource (see Table 6). For example, the property "nationality" in Figure 8 - and according to DBpedia - has Class "Person" as a domain and Class "Country" as a range. In the first step, the resource is identified using the NER system as mentioned in the resource identification section. As discussed earlier, extracted Named Entities represent the subject and/or object of the RDF triple and are used to extract possible properties based on SPARQL tables templates presented in Table 6. The retrieved list of properties is then matched with the expected range and domain lists, see Tables 5 and 6. To explain the process, Figure 7 presents an example of property identification of the question "ما هي عاصمة الاردن؟/mA hy EASmp AlArdn?/what is the capital of Jordan?". First, the resource "http://dbpedia.org/resource/Jordan" is obtained in the resource identification process. In the next step, the expected domain and the

TABLE 6. SPARQL query templates for property identification.

SPARQL template for domain extraction	SELECT DISTINCT ?d WHERE {resource rdf:type ?d .}
SPARQL query to identify property using expected range and domain.	PREFIX rdfs:<http://www.w3.org/2000/01/rdf schema#> prefix xsd:<http://www.w3.org/2001/XMLSch ema#> SELECT DISTINCT ?p WHERE { {Resource ?p ?y.} union {?y ?p Resource.} {?p rdfs:domain ?x. FILTER (?x IN (domain)).} {?p rdfs:range ?x2. FILTER (?x2 IN (range)).}}

expected range are extracted. After that, the expected domain and range are used to construct a SPARQL query that retrieves the "http://dbpedia.org/ontology/capital" as a property of the RDF triple.

This model relies on pattern matching for all possible RDF triples of the extracted Named Entities out of the question, and their possible properties. The approach is time-consuming as it traverses all the possible graphs attached to a specific Subject and/or Object (based on extracted Named Entities), and may lead to incorrect answers.

2) WIKIDATA-BASED APPROACH

Wikidata is a community-created knowledge base for Wikipedia launched in October 2012. The data in Wikidata are highly interlinked and connected to many other LOD graphs such as DBpedia. Every page in Wikidata is called an entity and has two main types: items and properties. Our interest is focused on the property type. Every property page contains descriptive data about the property including labels and datatype [47]. Property entity has a label for different languages including the Arabic language. These labels are used to address the challenge of the DBpedia lexical gap between Arabic question keyword and DBpedia property labels.

In this approach, property linking is done using the following steps: firstly, we extract the candidate keywords from the input question by removing the extracted named entities and Arabic stop words. The rest of the question's tokens are used to extract candidate properties using Wikidata graphs. Secondly, equivalent properties for the DBpedia are extracted form Wikidata using the SPARQL query that contains (owl: equivalentProperty) as presented in Table 7-first row. The Subject in the SPARQL template is replaced with the Named Entities extracted from the input question. Third, a SPARQL query that contains (rdfs:label and wikibase:label) is sent to Wikidata endpoint (https://query.wikidata.org/) to return the Arabic label for the candidate property (see Table 7-second row). After that, the list of tokens extracted in step 1 is matched with the Wikidata-based labels to retrieve the most similar label. Both the list of tokens and the retrieved Wikidata properties' labels are stemmed to facilitate the matching process. In case the matching process does not return any result; the list of Arabic labels is extended with their

Question in Arabic: ما هي عاصمة الأردن؟
Buckwalter Transliteration: mA hy EASmp AlArdn?
Question in English: What is the capital of Jordan?
The identified resource: <http://dbpedia.org/resource/Jordan >
The expected domain:
http://dbpedia.org/ontology/Place
http://dbpedia.org/ontology/Location
http://dbpedia.org/ontology/Country
http://dbpedia.org/ontology/PopulatedPlace
The expected range:
http://dbpedia.org/ontology/Location
http://dbpedia.org/ontology/City
http://dbpedia.org/ontology/PopulatedPlace
http://dbpedia.org/ontology/Place
http://dbpedia.org/ontology/Country
http://dbpedia.org/ontology/Organization
SPARQL query for property identification:

```
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
prefix xsd:<http://www.w3.org/2001/XMLSchema#>
SELECT DISTINCT ?p WHERE{{<http://dbpedia.org/resource/Jordan> ?p ?y.}
 union {?y ?p <http://dbpedia.org/resource/Jordan>.}
{?p rdfs:domain ?x.
FILTER (?x IN (<http://dbpedia.org/ontology/ Place >, <http://dbpedia.org/ontology/ Location >,
<http://dbpedia.org/ontology/ Country>, < http://dbpedia.org/ontology/PopulatedPlace>)).}
{?p rdfs:range ?x2.
FILTER (?x2 IN (<http://dbpedia.org/ontology/Location>,<http://dbpedia.org/ontology/City>,
<http://dbpedia.org/ontology/PopulatedPlace>,<http://dbpedia.org/ontology/Place>,
<http://dbpedia.org/ontology/Country>,<http://dbpedia.org/ontology/Organization>)).}}
```

The identified DBpedia Property: <http://dbpedia.org/ontology/capital>

FIGURE 7. An example of property identification of the question "ما هي عاصمة الاردن؟/mA hy EASmp AlArdn?/what is the capital of Jordan?".

synonyms using "skos: altLabel" that returns the alternative lexical label as presented in Table 7-third row.

Table 8 presents property identification process using the Wikidata-based approach for the question "ماهي سويدان؟ التي تخرج منها طارق الجامعة/mA hy AljAmEp Alty txrj mnhA TArq swydAn/ What university did Tareq Al-Suwaidan graduated from?". As presented in the example, the word الجامعة/AljAmEp/university from the list of remaining tokens has a synonym تعلم في which is the Arabic label for the Wikidata property P69 (http://www.wikidata.org/entity/P69).

3) DEPENDENCY PARSING APPROACH

A dependency parsing tree of a sentence is an acyclic directed graph that represents grammatical relations between nodes and edges [48]. A dependency parser expresses the grammatical relations between two nodes as a triple (head, relation, dependent), where a head is a word modified by the dependent. In this approach the Stanford Parser [49] is used. Stanford Parser is a natural language parser created by the Stanford Natural Language Processing Group. All possible grammatical relations generated by Stanford Parser have a description in the Stanford type dependencies manual [49].

Based on dependency parsing, the head-relation-dependent triple is used to extract the RDF triple. In the first step, we extract the Named Entities of the question then identify their DBpedia corresponding resources. After that, the dependency parsing process is applied to the input question. Thereafter, the extracted Named Entities which represent the head are used to extract the dependent keyword using the

TABLE 7. Wikidata-based SPARQL query templates.

SPARQL query to return pairs of DBpedia property and the equivalent Wikidata property	SELECT DISTINCT ?property ?x WHERE { Subject ?property ?y. ?property owl:equivalentProperty ?x . }
SPARQL query to return the label of Wikidata property	SELECT ?property ?propertyLabel ?LabelArabic WHERE { ?property a wikibase:Property . bind (< *property* > as ?property). ?property rdfs:label ?LabelArabic. filter (lang(?LabelArabic) ="ar"). SERVICE wikibase:label {bd:serviceParam wikibase:language "en" .}
SPARQL query to return the synonyms for the Wikidata property	PREFIX rdfs:<http://www.w3.org/2000/01/ rdf-schema#> SELECT ?property ?propertyLabel ?x (GROUP_CONCAT(DISTINCT(?altLabel); separator = ", ") AS ?altLabel_list) WHERE { ?property a wikibase:Property . bind (< *property* > as ?property). OPTIONAL { ?property skos:altLabel ?altLabel. FILTER (lang(?altLabel) = "ar") } ?property rdfs:label ?x. filter (lang(?x) = "ar"). SERVICE wikibase:label { bd:serviceParam wikibase:language "en" .} } GROUP BY ?property ?propertyLabel ?x

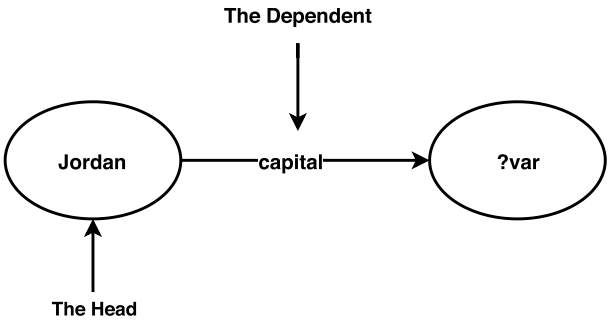

FIGURE 8. The head and dependent for "ما هي عاصمة الاردن؟/mA hy EASmp AlArdn?/what is the capital of Jordan?".

dependency parsing head-relation-dependent triple. As depicted in Figure 8, the head word "الاردن/Jordan" is modified by the dependant word "عاصمة/Capital". Finally, we use the extracted dependent word as a candidate property to identify the DBpedia corresponding property based on the steps explained earlier in the Wikidata-based approach.

C. QUERY FORMULATION

Extracted named entities and properties based on previous steps and processes are then used to formulate the SPARQL queries to retrieve the answer. This is done based on the formulation of the RDF triplets of the extracted data. In this

TABLE 8. Property identification using Wikidata-based approach for "ما هي الجامعة التي تخرج منها طارق سويدان؟/mA hy AljAmEp Alty txrj mnhA TArq swydAn/What university did Tareq Al-Suwaidan graduated from?".

Question	ما هي الجامعة التي تخرج منها طارق سويدان؟/mA hy AljAmEp Alty txrj mnhA TArq swydAn/What university did Tareq Al-Suwaidan graduated from?
The resource	http://dbpedia.org/resource/Tareq_Al-Suwaidan
Candidate Keywords	{الجامعة/AljAmEp/university تخرج/txrj/graduated}
DBpedia property list with corresponding Wikidata property	http://dbpedia.org/ontology/birthPlace= http://www.wikidata.org/entity/P19 http://dbpedia.org/ontology/birthDate= http://www.wikidata.org/entity/P569 http://dbpedia.org/ontology/birthYear= http://www.wikidata.org/entity/P569 http://dbpedia.org/ontology/education= http://www.wikidata.org/entity/P69 http://dbpedia.org/ontology/religion= http://www.wikidata.org/entity/P140
Wikidata properties Arabic label	مكان الولادة=http://www.wikidata.org/entity/-P19, تاريخ الميلاد =http://www.wikidata.org/entity/-P569, الدين =http://www.wikidata.org/entity/P140, تعلم في=http://www.wikidata.org/entity/P69
Synonym list	تعلم في ={ جامعة , تعلم في, جامعة ام, مدرسة ام } الدين=الدين
The Wikidata property	http://www.wikidata.org/entity/P69
The DBpedia property	http://dbpedia.org/ontology/education

research, the RDF triples are classified into two categories: (a) subject-based triple and (b) object-based triple. The subject-based triple indicates that the subject and predicate were successfully extracted, whereas the object is missing. Example on the subject-based triple is the triple extracted for the question "ما هي عاصمة سوريا؟/mA hy EASmp swryA?/what is the capital of Syria?":

- SUBJECT:< *http* : *//dbpedia.org/resource/Syria* >
- PROPERTY:< *http* : *//dbpedia.org/ontology/capital* >
- OBJECT:?*var*

In the object-based triple, the predicate and object are extracted, whereas, the subject of the RDF triple is missing; such as the triple extracted for the question "التي عاصمتها دمشق؟ما هي الدولة/mA hy Aldwlp Alty EASmthA dm$q?/what is the country that Damascus is the capital of?":

- SUBJECT:?*var*
- PROPERTY:< *http* : *//dbpedia.org/ontology/capital* >
- OBJECT:< *http* : *//dbpedia.org/resource/Syria* >

This research does not focus on triples where the relation is missing. For instance in a question like "what is Cairo to Egypt?" the question may have different answers depending on the RDF triple connecting the "subject Cairo" to the "object Egypt", or vice versa. This question may have answers like "Cairo is the largest city in Egypt", "Cairo is the Capital city of Egypt", or "Cairo is located in Egypt", etc. This type of questions needs further human interaction

TABLE 9. Basic SPARQL query templates.

Question Category	Template
General template	PREFIX rdfs:<http://www.w3.org/2000/01 /rdf-schema#> SELECT DISTINCT ?answer WHERE { subject property ?object. ? object rdfs:label ? answer. FILTER(LANG(?answer)='ar') }
Definition template	PREFIX rdfs:<http://www.w3.org/2000/01 /rdf-schema#> SELECT DISTINCT ?answer WHERE{ subject <http://dbpedia.org/ontology/ abstract> ? answer. FILTER(LANG(?answer)='ar') .}
List template	PREFIX rdfs:<http://www.w3.org/2000/01 /rdf-schema#> SELECT DISTINCT ?answer WHERE { subject property ?object. ?object rdfs:label ? answer. FILTER(LANG(?answer)='ar') . Limit no }
Count template	SELECT distinct (count(?answer) AS ?count) WHERE { s o ?answer. }
Yes/no template	ASK Where {subject property object. }

to select which answer is correct, and in question answering systems the challenge is to come up with a correct single answer for the user's question, with the very minimal human effort.

The final SPARQL query is generated using the extracted RDF triple using a template-based approach [20] of SPARQL queries for each type of the user's questions (see Table 9). The factoid, Definitional and List question requires the SELECT block to construct a SPARQL query, while the yes/no question requires the ASK block to construct a SPARQL query.

D. ANSWER EXTRACTION AND VALIDATION

In this phase, the SPARQL query is ready to be sent to the DBpedia endpoint and fetch the final answer. Later, the answers are evaluated using the gold-test set, and the Precision, Recall, and F-measure matrix are computed.

V. EXPERIMENTATION AND RESULTS

The proposed approaches were evaluated using our constructed dataset (i.e. 400 question). Results are reported using the Precision, Recall, F-Measure values.

A. EVALUATION METRICS

- **Precision** Precision (P) measures the ratio of number of correctly answered questions to the total number of answered questions. The precision is calculated by applying the following formula [50].

$$P = \frac{number\ of\ correctly\ answered\ questions}{total\ number\ of\ answered\ questions}$$

- **Recall** Recall (R) measures the ratio of the number of correct answers returned by the system to the number of gold standard answers. The recall is calculated by applying the following formula [50].

$$R = \frac{number\ of\ correctly\ answered\ question}{total\ number\ of\ question}$$

- **F-measure** The F-measure (F) is the weighted average of Precision and Recall. F-measure values ranges from 0 to 1, with 1 indicating the best score, and 0 indicates the worst. The F-measure is calculated using the following formula [50].

$$F = 2 * \frac{P * R}{P + R}$$

B. RESULTS AND FINDINGS

As presented in Table 10, the dependency parsing based approach overcame the other two approaches, and answered 385 questions out of 400 questions. A total of 325 questions out of the answered 385 questions were correct, and only 60 were answered incorrectly. Based on that, the model achieved good results with precision = 0.840, recall = 0.813, and F-measure = 0.828, with an enhancement over the baseline model by 21.7% with respect to the precision measure, 25.3% for the recall measure, and 24.8% for the F-measure.

The Wikidata-based approach came in the second level, and also achieved good results. The approach was able to answer 362 questions out of the 400 defined questions; 277 questions out of them were answered correctly, and 85 were answered incorrectly. This approach's results were as follows: precision = 0.765, recall = 0.692, and F-measure = 0.727.

Finally, the baseline approach came in third level and managed to answer 359 questions. A sum of 224 questions were answered correctly, and 135 were answered incorrectly. The approach's achieved a precision = 0.623, recall = 0.56, and F-measure = 0.58.

VI. DISCUSSION

After analyzing the results of the proposed Arabic QAS using LOD, and by comparing the results achieved by our proposed approaches to the related works implemented to participate in QALD 1-7 challenges, (see Section II, Subsection A), it is shown that our best performing approach (dependency parsing-based QAS) outperforms all the reported systems. Although the comparison might be subjective with respect to the difference in the datasets used by the different systems compared to our dataset, and the language of the questions

TABLE 10. Summary for the results of the proposed approach.

Approach	Number of the system answers	Number of questions answered correctly	Number of questions answered incorrectly	Precision	Recall	F-Measure
Baseline approach	359	224	135	0.623	0.56	0.58
Wikidata-based	362	277	85	0.765	0.692	0.727
Dependency parsing	385	325	60	0.840	0.813	0.828

(English vs. Arabic), the results are still very promising and unprecedented with respect to the difficulty of the task and the results obtained by other works (results achieved in terms of F-measure were below 80%).

Focusing on the limitations of the proposed approaches, the following list contains a summery of faced limitations in our proposed system:

1) Errors in the Named Entity Recognition process: The tool FARASA [46] tags only the basic classes (Person, Location, Organization). Moreover, in some cases, FARASA was not able to recognize the correct entity such as the person named entity "جمال سامية/sAmyp jmAl/Samia Gamal " in سامية جمال؟ما هي مهنة/mA hy mhnp sAmyp jmAl?/what is the occupation of samia gamal?" was recognized as "جمال"/jmAl/Beauty. In addition, some entities were recognized in a format that does not match the format of DBpedia resources, such as "اين يقع نهر النيل؟/Ayn yqE nhr Alnyl?/where is the nile river located?"; the named entity "نهر النيل/nhr Alnyl/Nile" was recognized as "النيل/Alnyl/Nile". Some other named entities, which have a certain level of ambiguity, such as the named entity "برشلونة/br$lwnh/FC Barcelona" in the question "ماهو لقب نادي برشلونة؟/mA hw lqb nAdy br$lwnh/", the system failed to tag the correct entity and tagged "Barcelona" the city instead of "FC Barcelona" the club. The limitations in FARASA tagging capabilities and the limitations in the named-entity disambiguation process (NER + Entity linking to DBpedia) led to having a failure in the resource identification process.
2) limitation in LOD (DBpedia) Arabic resources representation: the LOD, especially the DBpedia dataset, has a limitation in representing Arabic resources. The Arabic DBpedia chapter is not complete at the present, more work is needed to represent more Arabic resources and to include Arabic labels for available resources on the LOD and DBpedia.

On the approach-specific level; the baseline approach returned 224 correct answers out of 359 answered questions, which implies having considerable limitations (see Table 10).The limitations in this approach can be referred to the following reasons:

1) Error in the property identifying process: the identification process of the property depends on the expected range and domain. If the range or domain is an error or the domain and range are shared by multiple properties, then the returned property will not be correct. For example, in the question:" عملة السودان؟ما هي/mA hy Emlp AlswdAn?/what is the Sudan currency?", the extracted property must have returned "currency" as a property, but the baseline approach returned "capital" property as the "currency" property has a special range that is not supported by the baseline approach. The currency property has the range "http://dbpedia.org/ontology/currency" which is not supported by the system; as a consequence, the system failed to determine the correct range for that question.
2) The baseline approach was unable to decide which property is the correct if there are more than one property returned. For example: in the question "متى توفي ارسطو؟/mtY twfy ArwsTw?/when did Aristotle die?", the system returned "http://dbpedia.org/ontology/deathDate" and "http://dbpedia.org/ontology/birthDate" based on the range and the domain for the Named Entity Person (i.e. Aristotle). The system chose the property randomly from the list which was the wrong one "http://dbpedia.org/ontology/birthDate."

The Wikidata-based approach answered 277 questions correctly out of 362 questions that were answered by the system. This limitation is due to the fact that not all DBpedia properties have equivalent Wikidata property. For instance the DBpedia property "nationality" does not have an equivalent Wikidata property.

The dependency parsing approach was able to answer 387 questions out of 400 with F-measure of 78.7%. 310 questions were answered correctly and only 77 were answered incorrectly. Limitations in the model are due to the following:

1) limitations in the accuracy of the used dependency parser.
2) Property identification problem: as the dependency parsing-based model depends on the Wikidata-based model; the limitations discussed earlier for the wikidata-based model affected the results achieved in this model.

VII. CONCLUSION AND FUTURE WORK

In this work, (a) we have developed an Arabic Question Answering System based on Linked Open Data (LOD) to map Arabic questions into SPARQL queries and return answers from the DBpedia knowledge base. In addition, (b) we have

built an Arabic pair of Questions and Answers dataset with their SPARQL queries to evaluate our proposed models and to help researchers evaluate their models as well. Our system is divided into three main phases: (a) Question Analysis and Validation (b) Query Formulation, and (c) Answer Extraction phase. The question Analysis phase is the most important phase in our system. In this phase, we used different approaches to understand the intent of the user's question by mapping the question into SPARQL query with respect to DBpedia schema. The proposed approaches that are developed include (a) a baseline approach, (b) a Wikidata-based approach, and (c) a dependency parsing based approach.

To evaluate the performance of our proposed approaches, we used a 400 questions dataset with three main types of questions: Definition questions, Factoid questions and Yes/No questions. In addition, Precision, Recall, F-Measure are used as measures to show the efficiency of the proposed approaches in retrieving correct answers. Based on the evaluation results the dependency parsing based approach integrated with Wikidata-based approach achieved good results (precision = 84%, recall = 81.3%, and F-measure = 82.8%) with an enhancement over the baseline model by 21.7% for the precision measure, 25.3% for the recall measure, and 24.8% for the F-measure.

Results also show that the LOD (i.e. the DBpedia knowledge base in our case) has a limitation in answering Arabic questions until now and needs more effort to support the Arabic language. Our research integrates the DBpedia resource with other resources, such as Wikidata, to answer Arabic questions and to decrease the gap between the LOD vocabularies and the Arabic natural language vocabularies.

Leveraging LOD to automatically answer Arabic Questions is a challenging task, especially when it comes to facing the limitations in LOD representation of Arabic resources. Few Arabic knowledge bases are available on the LOD cloud and the knowledge base frequently used in QAS (i.e. DBpedia) has a limitation in representing Arabic web resources, or even providing Arabic labels to the represented resources.

In future work, we look forward to extending our system to answering additional categories of questions, such as causal and purpose questions. In addition, we plan to generate more patterns of SPARQL queries to cover complex questions such as the queries that contain aggregation function (i.e. "ORDER BY"). Moreover, we aspire to answer questions that contain more than one named entity and more than one property. Finally, to address the lexical gap between the Arabic language and the LOD terminology, we look to investigate lexical semantic similarity techniques such as using WordNet to get word synonyms.

REFERENCES

[1] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic Web," *Sci. Amer.*, vol. 284, no. 5, pp. 34–43, 2001.

[2] C. Bizer, T. Heath, D. Ayers, and Y. Raimond, "Interlinking open data on the Web," in *Proc. 4th Eur. Semantic Web Conf. Demonstrations Track*, Innsbruck, Austria, 2007, pp. 1–2.

[3] T. Heath and C. Bizer, "Linked data: Evolving the Web into a global data space," *Synth. Lect. Semantic Web, Theory Technol.*, vol. 1, no. 1, pp. 1–136, 2011.

[4] L. W. Lacy, *OWL: Representing Information Using the Web Ontology Language*. Bloomington, IN, USA: Trafford Publishing, 2005.

[5] J. G. Breslin, A. Passant, and S. Decker, "Interlinking online communities," in *The Social Semantic Web*. Berlin, Germany: Springer, 2009, pp. 197–250.

[6] D. Brickley, R. V. Guha, and B. McBride. (2004). *RDF Vocabulary Description Language 1.0: RDF Schema*. [Online]. Available: http://www.w3.org/tr/2004/rec-rdf-schema-20040210

[7] A. M. Ezzeldin and M. Shaheen, "A survey of arabic question answering: Challenges, tasks, approaches, tools, and future trends," in *Proc. 13th Int. Arab Conf. Inf. Technol. (ACIT)*, 2012, pp. 1–8.

[8] P. Rosso, Y. Benajiba, and A. Lyhyaoui, "Towards an Arabic question answering system," in *Proc. 4th Conf. Sci. Res. Outlook Technol. Develop. Arab World SROIV)*, Damascus, Syria, 2006, pp. 11–14.

[9] B. Gorenjak, M. Ferme, and M. Ojsteršek, "A question answering system on domain specific knowledge with semantic Web support," *Int. J. Comput.*, vol. 5, no. 2, pp. 141–148, 2011.

[10] O. Al-Qawasmeh, M. Al-Smadi, and N. Fraihat, "Arabic named entity disambiguation using linked open data," in *Proc. 7th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2016, pp. 333–338.

[11] J. Z. Pan and I. Horrocks, "RDFS(FA): Connecting RDF(S) and OWL DL," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 2, pp. 192–206, Feb. 2007.

[12] P. Gupta and V. Gupta, "A survey of text question answering techniques," *Int. J. Comput. Appl.*, vol. 53, no. 4, pp. 1–8, 2012.

[13] S. Kalaivani and K. Duraiswamy, "Comparison of question answering systems based on ontology and semantic Web in different environment," *J. Comput. Sci.*, to be published.

[14] I. Androutsopoulos, G. D. Ritchie, and P. Thanisch, "Natural language interfaces to databases—An introduction," *Natural Lang. Eng.*, vol. 1, no. 1, pp. 29–81, 1995.

[15] C. Unger, A. Freitas, and P. Cimiano, "An introduction to question answering over linked data," in *Reasoning Web. Reasoning on the Web in the Big Data Era*. Cham, Switzerland: Springer, 2014, pp. 100–140.

[16] L. Vanessa, C. Unger, P. Cimiano, and E. Motta, "Evaluating question answering over linked data," *J. Web Semantics*, vol. 21, pp. 3–13, Aug. 2013.

[17] S. K. Ray and K. Shaalan, "A review and future perspectives of arabic question answering systems," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3169–3190, Sep. 2016.

[18] D. Damljanovic, M. Agatonovic, and H. Cunningham, "FREyA: An interactive way of querying Linked Data using natural language," in *Proc. Extended Semantic Web Conf.* Berlin, Germany: Springer, 2011, pp. 125–138.

[19] V. Lopez, M. Fernández, E. Motta, and N. Stieler, "PowerAqua: Supporting users in querying and exploring the semantic Web," *Semantic Web*, vol. 3, no. 3, pp. 249–265, 2012.

[20] C. Unger, L. Bühmann, J. Lehmann, A.-C. N. Ngomo, D. Gerber, and P. Cimiano, "Template-based question answering over RDF data," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 639–648.

[21] A. Freitas and E. Curry, "Natural language queries over heterogeneous linked data graphs: A distributional-compositional semantics approach," in *Proc. 19th Int. Conf. Intell. User Interfaces*, 2014, pp. 279–288.

[22] N. Aggarwal and P. Buitelaar, "A system description of natural language query over DBpedia," in *Proc. Interact. Linked Data (ILD*, 2012, pp. 96–99.

[23] S. Walter, C. Unger, P. Cimiano, and D. Bár, "Evaluation of a layered approach to question answering over linked data," in *Proc. Int. Semantic Web Conf.* Berlin, Germany: Springer, 2012, pp. 362–374.

[24] E. Cabrio, J. Cojan, A. P. Aprosio, B. Magnini, A. Lavelli, and F. Gandon, "QAKiS: An open domain QA system based on relational patterns," in *Proc. Int. Semantic Web Conf. (ISWC)*, 2012, pp. 1–5.

[25] C. Pradel, O. Haemmerlé, and N. Hernandez, "A semantic Web interface using patterns: The SWIP system," in *Graph Structures for Knowledge Representation and Reasoning*. Berlin, Germany: Springer, 2012, pp. 172–187.

[26] K. Xu, S. Zhang, Y. Feng, and D. Zhao, "Answering natural language questions via phrasal semantic parsing," in *Natural Language Processing and Chinese Computing*. Berlin, Germany: Springer, 2014, pp. 333–344.

[27] L. Zou, R. Huang, H. Wang, J. X. Yu, W. He, and D. Zhao, "Natural language question answering over RDF: A graph data driven approach," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 313–324.
[28] S. Ruseti, A. Mirea, T. Rebedea, and S. Trausan-Matu, "Qanswer-enhanced entity matching for question answering over linked data," in *Proc. CLEF*, 2015, pp. 1–12.
[29] A. P. B. Veyseh, "Cross-lingual question answering using common semantic space," in *Proc. TextGraphs-10 Workshop Graph-Based Methods Natural Lang. Process.*, 2016, pp. 15–19.
[30] A. Marginean, "Question answering over biomedical linked data with grammatical framework," *Semantic Web*, vol. 8, no. 4, pp. 565–580, 2017.
[31] R. Usbeck, A.-C. N. Ngomo, B. Haarmann, A. Krithara, M. Röder, and G. Napolitano, "7th open challenge on question answering over linked data (QALD-7)," in *Proc. Semantic Web Eval. Challenge*. Cham, Switzerland: Springer, 2017, pp. 59–69.
[32] D. Sorokin and I. Gurevych, "End-to-end representation learning for question answering with weak supervision," in *Proc. Semantic Web Eval. Challenge*. Cham, Switzerland: Springer, 2017, pp. 70–83.
[33] D. Diefenbach, K. Singh, and P. Maret, "WDAqua-core1: A question answering service for rdf knowledge bases," in *Proc. Companion Web Conference*, 2018, pp. 1087–1091.
[34] F. Mohammed, K. Nasser, and H. M. Harb, "A knowledge based Arabic question answering system (AQAS)," *ACM SIGART Bull.*, vol. 4, no. 4, pp. 21–30, 1993.
[35] B. Hammo, H. Abu-Salem, and S. Lytinen, "QARAB: A question answering system to support the Arabic language," in *Proc. ACL Workshop Comput. Approaches Semitic Lang.*, 2002, pp. 1–11.
[36] Y. Benajiba, P. Rosso, and A. Lyhyaoui, "Implementation of the arabiqa question answering system's components," in *Proc. Workshop Arabic Natural Lang. Process. 2nd Inf. Commun. Technol. Int. Symp. (ICTIS)*, Fes, Morroco, Apr. 2007, pp. 3–5.
[37] W. Brini, M. Ellouze, O. Trigui, S. Mesfar, H. Belguith, and P. Rosso, "Factoid and definitional arabic question answering system," in *Proc. Post-NOOJ*, Tozeur, Tunisia, Jun. 2009, pp. 8–10.
[38] S. Bekhti, A. Rehman, M. Al-Harbi, and T. Saba, "AQUASYs: An arabic question-answering system based on extensive question analysis and answer relevance scoring," *Int. J. Acad. Res.*, vol. 3, no. 4, pp. 45–54, 2011.
[39] O. Trigui, L. H. Belguith, and P. Rosso, "DefArabicQA: Arabic definition question answering system," in *Proc. 7th LREC Workshop Lang. Resour. Hum. Lang. Technol. Semitic Lang.*, Valletta, Malta, 2010, pp. 40–45.
[40] M. Nabil, A. Abdelmegied, Y. Ayman, A. Fathy, G. Khairy, M. Yousri, N. M. El-Makky, and K. Nagi, "AlQuAnS–An Arabic language question answering system," in *Proc. KDIR*, 2017, pp. 144–154.
[41] I. Lahbari, S. O. El Alaoui, and K. A. Zidani, "Toward a new Arabic question answering system," *Int. Arab J. Inf. Technol.*, vol. 15, no. 3A, pp. 610–619, 2018.
[42] A. W. AbuTaha and I. M. Alagha, "An ontology-based arabic question answering system," M.S. thesis, Central Library, Islamic Univ. Gaza, Gaza, 2015. [Online]. Available: https://pdfs.semanticscholar.org/cb58/6e05ca3298fa633f461031e11716ba3b285c.pdf
[43] I. A. Agha and O. El-Radie, "Towards verbalizing SPARQL queries in Arabic," *Eng., Technol. Appl. Sci. Res.*, vol. 6, no. 2, pp. 937–944, 2016.
[44] I. AlAgha, "Using linguistic analysis to translate arabic natural language queries to SPARQL," 2015, *arXiv:1508.01447*. [Online]. Available: https://arxiv.org/abs/1508.01447
[45] M. Althobaiti, U. Kruschwitz, and M. Poesio, "AraNLP: A Java-based library for the processing of Arabic text," in *Proc. 9th Lang. Resour. Eval. Conf. (LREC)*. Reykjavík, Iceland: European Language Resources Association, 2014.
[46] A. Abdelali, K. Darwish, N. Durrani, and H. Mubarak, "Farasa: A fast and furious segmenter for arabic," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Demonstrations*, 2016, pp. 11–16.
[47] F. Erxleben, M. Günther, M. Krötzsch, J. Mendez, and D. Vrande ić, "Introducing Wikidata to the linked data Web," in *Proc. Int. Semantic Web Conf.* Cham, Switzerland: Springer, 2014, pp. 50–65.
[48] R. McDonald and F. Pereira, "Online learning of approximate dependency parsing algorithms," in *Proc. 11th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2006, pp. 1–8.
[49] M.-C. De Marneffe, T. Dozat, N. Silveira, K. Haverinen, F. Ginter, J. Nivre, and C. D. Manning, "Universal stanford dependencies: A cross-linguistic typology," in *Proc. LREC*, vol. 14, 2014, pp. 4585–4592.
[50] C. Unger, C. Forascu, V. Lopez, A.-C. N. Ngomo, E. Cabrio, P. Cimiano, and S. Walter, "Question answering over linked data (QALD-4)," in *Proc. Workshop Notes CLEF Conf.*, 2014, pp. 1–10.

MOHAMMAD AL-SMADI received the Ph.D. degree in computer science from the Graz University of Technology, in 2012. He is currently an Associate Professor with the Computer Science Department, Jordan University of Science and Technology. He has coauthored several technical articles in established journals and conferences in fields related to social and semantic computing, knowledge engineering, natural language processing, and technology enhanced learning. He is co-chairing many IEEE events such as OSNT, SNAMS, BDSN, iLearn, and many others.

ISLAM AL-DALABIH received the B.S. degree in computer science from Al al-Bayt University, Jordan, in 2014, and the M.S. degree in computer science from the Jordan University of Science and Technology, Jordan, in 2019. Since 2015, she has been working as an Instructional Computer Lab Supervisor with the Computer Science Department, Al al-Bayt University.

YASER JARARWEH received the Ph.D. degree in computer engineering from the University of Arizona, in 2010. He is currently a Professor of computer science with Duquesne University. He has coauthored many technical articles in established journals and conferences in fields related to cloud computing, HPC, SDN, and Big Data. He is chairing many IEEE events such as AICCSA, SDS, FMEC, ICICS, SNAMS, BDSN, IoTSMS, and many others. He has served as a Guest Editor for many special issues in different established journals. Also, he is the Steering Committee Chair of the IBM Cloud Academy Conference. He is an Associate Editor of the *Cluster Computing Journal* (Springer), the *Information Processing and Management*, and others.

PATRICK JUOLA received the Ph.D. degree in computer science from the University of Colorado at Boulder, in 1995. He is currently a Professor of computer science with Duquesne University, where he directs the Evaluating Variations in Language Laboratory. He specializes in the analysis of texts to determine their authorship, and is best known for his 2013 analysis of *The Cuckoo's Calling*, a detective novel he revealed to have been written by J. K. Rowling.