Towards Verbalizing SPARQL Queries in Arabic

Iyad Al Agha
Faculty of Information Technology
The Islamic University of Gaza
Gaza Strip, Palestine
ialagha@iugaza.edu.ps

Omar El-Radie
Faculty of Information Technology
The Islamic University of Gaza
Gaza Strip, Palestine
omar.elradie@gmail.com

***Abstract*—With the wide spread of Open Linked Data and Semantic Web technologies, a larger amount of data has been published on the Web in the RDF and OWL formats. This data can be queried using SPARQL, the Semantic Web Query Language. SPARQL cannot be understood by ordinary users and is not directly accessible to humans, and thus they will not be able to check whether the retrieved answers truly correspond to the intended information need. Driven by this challenge, natural language generation from SPARQL data has recently attracted a considerable attention. However, most existing solutions to verbalize SPARQL in natural language focused on English and Latin-based languages. Little effort has been made on the Arabic language which has different characteristics and morphology. This work aims to particularly help Arab users to perceive SPARQL queries on the Semantic Web by translating SPARQL to Arabic. It proposes an approach that gets a SPARQL query as an input and generates a query expressed in Arabic as an output. The translation process combines both morpho-syntactic analysis and language dependencies to generate a legible and understandable Arabic query. The approach was preliminary assessed with a sample query set, and results indicated that 75% of the queries were correctly translated into Arabic.**

Keywords-SPARQL; Natural Language Processing; Ontology; Morpho-syntactic features; Arabic

I. INTRODUCTION

The Semantic Web is the next generation of the current Web. With the dramatic growth of the Linked Data Web in the past few years, an increased amount of RDF data has been published as Linked Data. Intuitive ways of accessing Linked Data have become highly demanded as a growing number of applications rely on RDF data as well as on the W3C standard SPARQL for querying this data [3]. SPARQL is the query language for the Semantic Web. Along with RDF and OWL, it is one of three core technologies of the Semantic Web. SPARQL enables Web applications and agents to query the Web, similar to how the SQL is used to query database Tables [4].

While SPARQL has proven to be a powerful tool in the hands of expert users, it remains difficult to understand for ordinary users who have limited experience with the Semantic Web. Therefore, it is necessary to bridge the gap between the query language understood by semantic data backbends, i.e. SPARQL, and that of the end users. Existing research has approached this problem by proposing interfaces to translate SPARQL queries to natural languages [5]. They often take a SPARQL query as an input and produce a question expressed in natural language. These interfaces have benefits for both naïve and expert users: A naïve user is enabled to reliably understand the content of SPARQL queries. Expert users can assess the correctness of the SPARQL queries they construct by translating them back to natural language.

Arabic is one of the largest members of the Semitic language family and is spoken by nearly 500 million people worldwide [6]. It is one of the six official UN languages. Despite its cultural, religious, and political significance, Arabic has received comparatively little attention in modern computational linguistics. Towards supporting the Arabic use on the Semantic Web, this work proposes a generic approach to translate SPARQL to Arabic queries, thus enabling Arab users to understand SPARQL and RDF data without exposing the underlying complexity.

While few efforts [7, 8] presented approaches to translate SPARQL queries to natural language, most of these efforts focused on the English language. These approaches, however, cannot be directly used for the Arabic language as it has different characteristics and morphology. In this research, we propose a generic approach to verbalize SPARQL queries in Arabic. It exploits natural language processing and language dependencies to translate RDF triples of SPARQL into legible Arabic sentences. It also exploits morphological analysis to realize the Arabic sentences and make them easy to read and understand.

II. RELATED WORKS

Approaches to facilitate access to ontologies and OWL back-ends have been the concern of many research works [9-12]. These works provide interfaces to enable querying RDF content by inputting natural language queries. In contrast, little research has been conducted to investigate the other way around, which is the verbalization of SPARQL. Verbalizing SPARQL queries enables users to understand how the results have been retrieved and whether the right question has been asked to the queried knowledge base.

In the domain of English language, some efforts aimed to generate natural language representations of portions of ontologies [13, 14]. These efforts claimed that the natural language representation of ontologies can be more informative to non-expert users than graphical representations. Dong and Holder [15] proposed a system that uses an ontology skeleton and handcrafted templates to generate natural language from semantic graphs. Wilcock [16] presented a generic approach

for verbalizing OWL and DAML+OIL ontologies. Third et al. [17] presented a tool that generates easily-navigable English text from OWL input. Other works on generating textual representations of OWL ontologies include [18-20].

Besides the work on OWL, research on generating textual descriptions of RDF triples has gained a considerable attention [5, 21, 22]. These solutions have been widely used for question answering over linked data and RDF back-ends [3, 23]. In addition, some approaches worked on the verbalization of first-order logics [24].

Some works have explored ways to translate database queries, i.e. SQL queries, to natural language text [19, 25, 26]. To the best of our knowledge, only two approaches were proposed for verbalizing SPARQL queries in English, which are SPARTIQULATION [8] and SPARQL2NL [7]. SPARTIQULATION is based on identifying the main entity in a SPARQL query. The main entity is then used to subdivide the query graph into sub-graphs that are ordered and matched with pre-defined message types. SPARTIQULATION is limited to SELECT queries and cannot handle important SPARQL features such as UNION and GROUP BY constructs. SPARQL2NL aims to generate English sentences by first generating a list of dependency trees for an input query, and then it applies reduction and replacement rules to improve the legibility of the verbalization. Unlike SPARTIQULATION, SPARQL2NL covers a wide range of SPARQL features and supports better realization of the sentence structure.

All the previous approaches exploited linguistic features and language patterns to generate natural language from queries. They benefited from the advancements in English NLP toolkits. However, it is difficult to generalize these approaches to the Arabic language. This is due to the fact that Arabic has more complex morphological, grammatical and semantic structures that make existing NLP techniques used for English inadequate for Arabic.

Building Arabic ontologies that can be used in a wide context is gaining momentum [6, 27, 28]. In addition, several ontology-based approaches have been proposed for enhancing information retrieval from the Arabic content on the Web [29, 30]. In line with these efforts, natural language interfaces will be demanded to enable naïve Arab users to send queries and obtain results from ontologies. To address this demand, only few efforts have explored the translation of queries expressed in Arabic to SPARQL [2, 31] To our knowledge, no work has explored the other direction, which is the verbalization of SPARQL queries in Arabic.

III. THE KNOWLEDGE BASE

Before presenting the approach for translating SPARQL to Arabic, it is necessary to give an overview of the knowledge base and the ontology used in the examples discussed in this paper. We used the Diseases ontology that was used in previous research [32]. The ontology is built in OWL and formally covers and classifies a range of well-known diseases in both English and Arabic. The ontology models the relationships between diseases, cures, symptoms, diagnoses and the organs of human body. An excerpt of the ontology schema including ontology classes (e.g. Cure, Disease, Symptom, Organ and Diagnosis) and the relationships between them, i.e. the object properties can be found in [2]. The ontology was populated with 149 ontology instances of different types. Table I gives information about the ontology content and size.

TABLE I. THE CONTENTS OF THE DISEASES ONTOLOGY

Statistics on the Diseases Ontology	
Number of Classes	12
Number of Object Properties	9
Number of Data Properties	1
Number of Instances	149

When translating SPARQL to Arabic, we assume that all terms of the input query should map to terms in the ontology. We also assume that the Arabic translations of query terms are available either from the ontology or from a dictionary. Starting with Arabic words that correspond to SPARQL terms, our approach focuses on exploiting linguistic analysis and natural language processing to link and inflect these words, and then improve the realization of the generated sentence. In this work, the ontology was populated with the Arabic translations of all ontology concepts, properties and instances, whereas a property (rdfs:label) is used to maintain the translations. These translations were retrieved from the ontology and used during the verbalization process.

IV. THE APPROACH TO VERBALISE SPARQL IN ARABIC

The proposed approach for translating SPARQL queries to Arabic consists of the four consequent stages shown in Figure 1. Each stage has an input taken from an earlier stage, and produces an output that is fed into the following stage. These stages are: query preprocessing, converting SPARQL to Arabic sentences, sentence realization and post-processing. Each stage is explained in what follows.

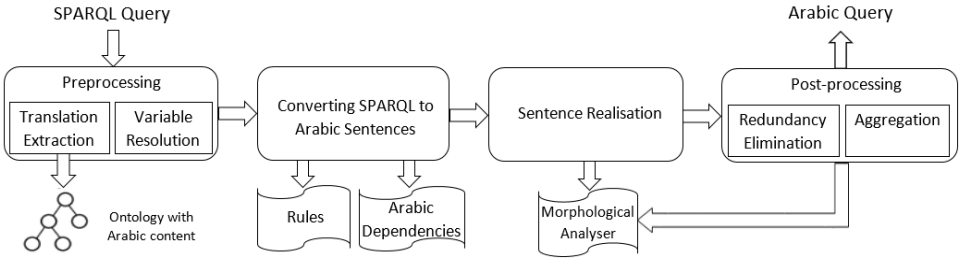

Fig. 1. The stages of the SPARQL to Arabic translation approach

A. SPARQL Pre-processing

The pre-processing stage aims to replace each term in the input SPARQL query with its corresponding translation in Arabic. As mentioned earlier, we assume that the Arabic translations of ontology terms are stored and can be retrieved from the ontology by referring to the property rdfs:label.

A SPARQL query typically consists of one or more triples with one or more variables. A variable in SPARQL can be a subject, a predicate or an object. These variables often refer to the target words in the natural language query. When translating SPARQL to natural language, each variable in SPARQL should be replaced by an Arabic word indicating its type, i.e. the ontology entity it refers to. Types of variables will be used as the target words in the generated Arabic question. For example, the query: "SELECT ?x WHERE {?x :cures :Diabetes}" has one variable, i.e. ?x, which denotes the medication or the cure of diabetes. Referring to the schema in [2], the variable ?x in the former query is of type :Cure which translates into Arabic as "علاج". This Arabic translation is then used to formulate the Arabic question by appending it to the question word as the following: "... ما علاج".

Let ?x be a variable in SPARQL query, and C the set of ontology classes. To determine the type of ?x, we first search the input query for the graph patterns: "?x rdf:type c" where c $c \in C$. If it exists, the Arabic translation of the class c is retrieved from ontology by referring to the property rdfs:label. If the type cannot be explicitly determined from the query, then it can be identified by matching the query triples with the ontology content. For example, the triple <?x :cures :Diabetes> matches with the ontology triple: :Insulin_tips :cures :Diabetes. Then, the type of the variable ?x is the direct class to which the instance :Insulin_tips belongs.

The output of this step is a sequence of Arabic words that correspond to the SPARQL terms. For example, if the input query is "SELECT ?x WHERE {?x :has_symptom :High_Blood_Pressure . ?x :infects :Lungs}", then the output is the following sequence (from left to right): "الرئتان<< يصيب <<مرض . ارتفاع ضغط الدم <<له_عرض<<مرض". Note that this step does not consider ordering and inflecting the generated words to produce a syntactically-valid sentence. The syntactic and morphological realization of the sentence will be handled in the subsequent steps.

B. Decomposing the UNION construct

Part of the pre-processing step is to handle the UNION operator that may be part of the SPARQL query. The UNION operator is used to link multiple triples in SPARQL, and translates into natural language as a disjunction, i.e. "أو". For example, the query "SELECT ?x where {{?x :has_symptom :Irritation}UNION{?x :has_symptom :Fever}}" is translated as "ما المراض الذي من أعراضه ارتفاع ضغط الدم أو الحمى؟". Triples linked by the UNION operator should be split so that they can be processed and translated separately. A complete Arabic statement is then generated by joining the statements corresponding to triples by disjunctions (أو).

C. Arabic Language Dependencies

The Arabic words generated from the pre-processing stage should be linked together to formulate a well-structured sentence. Relationships between words in a sentence are typically captured from a dependency tree constructed by a parser. In this work, we need to perform the opposite task: Starting by standalone words, we need to link and inflect these words by choosing appropriate relationships that yield a valid sentence.

Existing efforts in the domain of English text [7] have often relied on language dependencies, such as Stanford Language Dependencies [33], to identify relations between words. However, English dependencies cannot be directly used for the Arabic language which uses different structures. Therefore, we started with Stanford Dependencies and tried to adapt them for the Arabic language. We defined a set of Arabic language dependencies that resemble the structures of simple Arabic sentences. These dependencies were defined by referring to similar dependencies in English and modifying them to cope with the Arabic language. Each dependency defines a relationship between words that occur in the text. For example, the dependency subj(مرض, يصيب) refers to the relationship between a subject and a verb, obj(ينكرياس, يصيب) refers to the relationship between a verb and an object. Note that these dependencies do not cover all possible structures of Arabic sentences. It only covers the basic structures enough to handle non-complex SPARQL queries.

TABLE II. ARABIC DEPENDENCES

Dependency	Explanation
nn	A noun compound modifier: a head noun is modified by another noun. For instance: nn(مرض, السكر) stands for مرض السكر.
subj	A dependency between a subject and a verb, for example subj(أحمد , أكل) expresses أحمد أكل.
obj	A verb phrase dependency between a verb and its direct object, for example obj(أكل,التفاحة) expresses أكل التفاحة.
amod	An adjectival modifier of a noun phrase is any adjectival phrase that serves to modify the meaning of the noun phrase. Represents the adjectival modifier dependency. For example, amod(الغدد,الصماء)stands for الغدد الصماء .
neg	Negation modifier: The negation modifier is the relation between a negative word and the word it modifies. For example neg(لا،يحتوي)هذا الكتاب لا يحتوي على تاريخ الاندلس
pron	Subject Pronouns: defines a relation between a pronoun and a subject. For example, pron(هو, علاج) expresses هو علاج.
poss	Expresses a possessive dependency between two lexical items, for example poss(أحمد, كتاب) expresses أحمد كتاب.

D. Converting SPARQL Triples to Arabic Sentences

After defining a set of Arabic language dependencies, we defined a set of rules for mapping SPARQL triples to Arabic sentences. These rules determine what dependencies should be used based on the structure of each triple. Formally, let f be an interpretation function that takes the Arabic words, generated from the pre-processing stage, and generates a structured Arabic sentence. The rules are as the following:

Rule 1: f (s p o) where p is a verb and s is a variable => subj (p, s) ^ obj(p, o)

where s, p and o stand for the Arabic words that correspond to the subject, the predicate and the object of the triple respectively. This rule means that if the predicate is a verb and the subject is a variable, then it is transformed to a natural language by using a concatenation of subj, obj dependencies from Table II.

Given the following sample triple: <?var :infects :organ>, the subject is unknown, and the predicate :infects refers to the verb "يصيب". The type of the variable ?var is :Disease. On applying Rule1, we get:

f (s p o) => *f* (<:Disease :infects :Organ>)
f (s p o) => subj(:Disease :infects) ^ obj(:infects :Organ)
f (s p o) => مرض يصيب ^ يصيب عضو
f (s p o) => مرض يصيب عضو (After removing redundancy).

Rule 2: f (s p o) where p is a noun => poss(p, o) ^ pron (s)

This rule indicates that if the predicate is a noun, then the poss dependency is used to link the predicate and the subject. The generated clause is then linked with the object by using the pron dependency. For example, in the triple: <:Headache :symptom_of :Sinus _Infection>, the predicate :symptom_of refers to the noun "عرض". On applying rule 2, we get:

f(s p o) =>*f* (<:Headache :symptom_of :Sinus_Infection>)

f (s p o) => poss(عرض، التهاب الجيوب الأنفية) ^ pron(هو, الصداع)

f (s p o) => هو الصداع ^ عرض التهاب الجيوب الأنفية

f (s p o) => عرض التهاب الجيوب الأنفية هو الصداع

Rule 3: *f* (s p o) where p is rdf:type => nn(s ,o)

This rule indicates that if the predicate is rdf:type, the nn dependency is used to link the subject with the object. For example, given the triple <:Diabetes rdf:type :Disease>, Rule 3 is applied as the following:

f (s p o) =>*f* (:Diabetes rdf:type :Disease) => nn(:Diabetes, :Disease) => مرض السكر

E. Sentence Realization

The rules in the previous section define how to link the Arabic words that correspond to the SPARQL terms. However, there are still several challenges that should be considered to support the realization of the generated verbalization. This may require inflecting, reordering or putting words in other formats. In the following we discuss these challenges:

- *Passive or active voice*: The voice of generated sentences sometimes should to be changed from active to passive and vice versa. The correct voice to use depends on the variables in the SPARQL query, and whether these variables map to objects or subjects. The intention to change the sentence's voice is to bring the target words, which correspond to the SPARQL variables, to the beginning of the sentence so that they precede other words. This will simplify the construction of the natural language query. Take the following query as an example: SELECT ?x WHERE {:Diabetes :infects ?x}. According to Rule 1, the triple in the former query is translated as " السكر يصيب عضو", where the word "عضو" is the type of the variable ?x. The word "عضو" is also the object and the target of the query. Thus, the triple should be revised so that the object becomes the subject and vice versa. This means that the sentence's voice should be changed from active to passive to become: "عضو يصاب بالسكر". Changing the sentence's voice brings the target word, i.e. "عضو" to the beginning, thus allowing to easily construct the Arabic question by adding the appropriate question word before the target word, i.e. " ما العضو الذي يصاب بالسكر؟".

- *The gender (masculine or feminine)*: Sometimes it is necessary to reform Arabic words in accordance with the intended gender. For example, the intention of the query SELECT ?x WHERE { :Bacteria :causes ?x } is to retrieve the diseases caused by bacteria. The Arabic representation of this query should look like: "ما المرض الذي تسببه البكتيريا؟". Note that the suffix of the verb "تسببه" as well as the pronoun "الذي" both refer to a masculine noun which is "المرض".

- *Singular and plural nouns:* As with the gender of nouns, singular and plural nouns should also be considered when inflecting the Arabic words. Note that an Arabic noun may have regular or irregular plurals. The plural words may also sound feminine, i.e."الأعراض" or masculine, i.e. "مخترعون". Note also that using singular or plural formats can modify the formats of other words in the sentence such as pronouns and prefixes. For example, the plural noun "أعراض" is used with the pronoun "التي", while the singular noun "عرض" is used with the pronoun "الذي".

To address the above linguistic challenges, the Arabic sentence generated from the previous step is processed by means of linguistic analysis. For this purpose, we used the Arabic Toolkit Service (ATKS) [1] which is a set of NLP components developed by Microsoft and targeting the Arabic language. Among its various components, ATKS includes a full-fledged morphological analyzer that supports a variety of features such as word synthesis, part of speech tagging and generation of derivatives.

We used a set of features suggested by ATKS morphological analyzer to improve the legibility of the Arabic query. For each Arabic word inputted to the analyzer, it generates a set of morpho-syntactic features. Among many things, these features include the voice of the verb, i.e. " معلوم أو مجهول", the gender of nouns and the number, i.e. singular or plural. We rely on these features to identify the status of Arabic words and modify them if necessary. For example, a sentence can be changed from active to passive voice using the following steps: 1) the morphological analyzer is used to identify the status of the verb in the sentence, i.e. active or passive. 2) All possible derivatives of the verb are generated with the help of the morphological analyzer. 3) These derivatives are then searched for a verb that has the same grammatical tense as the original verb but has an opposite voice. 4) The voice of the sentence is finally changed by swapping the subject and the object, and replacing the verb with the derivative identified from step 3. Figure 2 shows the

pseudo code of the algorithm used for changing the voice of a sentence.

Input: A sentence S = <s, p, o> where s denotes a subject, o denotes an object, and p denotes a predicate.
Output: A sentence $\overline{S}$ = <o, $\overline{p}$, s> where the voice of $\overline{S}$ is the opposite of the voice of S .
Use Variables:
F1, F2: empty lists of morpho-syntactic features
D: An empty list of derivatives
V1, V2: the voice of word (active or passive)
Begin
// Analyze p using ATKS morphological analyzer
F1:= getMorpho_syntactic_features(p)
// Get the voice feature from the features list
V1:= getVoice(F1)
// Get all derivatives of p using ATKS morphological analyzer
D := getDerivatives(p)
for each d in D **do**
F2 := getMorpho_syntactic_features(d)
V2 := getVoice (F2);
// If the original predicate and its derivative have that same tense but different voices
If (tense(V1) == tense(V2) AND V1 != V2) **then**
$\overline{p}$:= p
Break
End if
End for
$\overline{S}$:= <o, $\overline{p}$, s>
End

Fig. 2. The algorithm of converting the sentence's voice from active to passive and vice versa.

The gender and number of Arabic words can be similarly determined by morpho-syntac features. As for the output of the morphological analysis, each word is tagged as "Masculine" or "Feminine" words, and is tagged as "Singular" or "Plural" (see Figure 3 for an example). If there is a need to invert the gender, or change the number, we similarly generate all possible derivatives of the word, and then choose the derivative words that fulfil the translation needs.

F. Aggregation and Post-processing

After identifying the proper voice, gender and number, the Arabic query is constructed. So far, each SPARQL triple is translated to a single Arabic sentence. The next step is to aggregate these sentences to construct the target query in natural language. The aggregation process typically involves three steps that are illustrated in the pseudo code in Figure 3: 1) Redundant mentions of variables are removed. If redundant variables denote nouns, they are replaced by proper pronouns to preserve the coherence of the sentence. These pronouns are chosen based on the morpho-syntactic features of the noun, i.e. the gender and the number. 2) Disjunctions, i.e. "أو" or conjunctions, i.e. "و" are used to link sentences. 3) An appropriate question word is added to generate the query.

The following example illustrates how the aggregation process is carried out: the query: SELECT ?x where {?x :diagnosed_by :CT_Scan . :Surgery :cures ?x is translated as "مرض يشخص بالأشعة المقطعية . مرض يعالج بالجراحة". This output, which consists of two separate sentences, is generated after applying the dependency rules and the morphological analysis as explained in the earlier sections. The two sentences are aggregated as the following: First, redundancy is eliminated by replacing the second occurrence of "مرض" with the pronoun "هو". Then, a conjunction is used to link the two sentences (A disjunction to be used if UNION operator is used). This results in the sentence: "مرض يشخص بالأشعة المقطعية وهو يعالج بالجراحة". Finally, the question word "ما" is inserted at the beginning of the sentence to formulate the final query: "ما مرض يشخص بالأشعة المقطعية وهو يعالج بالجراحة ؟".

Input: A sequence of Arabic sentences S that correspond to RDF triples, A connector c ∈ {"و","أو"}
Output: An Arabic query q
Begin
For each sentence s ∈ S
For each word w ∈ s
If w is redundant **then**
If w is a noun **then**
Replace w with a pronoun
Else
Remove w
End if
End if
End for
If q is empty **then**
q := s
Else
q := c ^ s
End if
End for
q := getQuestionWord(q) ^ q
End

Fig. 3. The algorithm of converting the sentence's voice from active to passive and vice versa.

V. A COMPLETE EXAMPLE

In the following, a complete example is presented to illustrate how the proposed approach for verbalizing SPARQL in Arabic is applied. Assume that the input query is:

SELECT ?x where {?x :infects :Pancreas . :Insulin :cures ?x}. This query has one variable, ?x, and two triples. The query aims to retrieve diseases that infect the Pancreas and are treated by Insulin. The verbalization process is carried out as the following:

Step 1: Arabic translations of query terms are extracted from the ontology: This generates the following sequence:

"?x >> يصيب >> البنكرياس . >> الأنسولين >> يعالج >> ?x".

Step 2: Variable are replaced by their corresponding type. The query does not contain an rdf:type triple. Therefore, the type of ?x can be identified by matching the query triples with the

ontology content. This can be achieved by executing the following query over the ontology:

SELECT ?type WHERE {?x rdf:type ?type . ?x :infects :Pancreas . :Insulin :cures ?x}

This query retrieves the type :Disease which is translated as "مرض". Finally, the mentions of variable ?x are replaced by the word "مرض", resulting in the following sequence:

"مرض << يعالج << الأنسولين . << البنكرياس << يصيب << مرض".

Step 3: Arabic sentences are constructed by applying the appropriate dependency rules. Both triples have predicates that are verbs. Then, rule 1 applies as the following:

For the first triple:
f (s p o) => *f* (<:Disease :infects :Pancreas>)
f (s p o) => subj(:Disease :infects) ^ obj(:infects :Pancreas)
f (s p o) => مرض يصيب ^ يصيب البنكرياس
f (s p o) => مرض يصيب البنكرياس (After removing redundancy).

For the second triple
f (s p o) => *f* (<:Insulin :cures :Disease>)
f (s p o) => subj(:Insulin :cures) ^ obj(:cures :Disease)
f (s p o) => الإنسولين يعالج ^ يعالج مرض
f (s p o) => الإنسولين يعالج مرض (After removing redundancy).

Step 4: Morphological analysis is performed to determine the correct voice, gender and number. This results in the following sentences: مرض يصيب البنكرياس . مرض يعالج بالإنسلوين. Note that the voice of the latter sentence is changed from active voice to passive in order to bring the word "مرض" to the beginning of the sentence.

Step 5: Redundant words are eliminated, and a question word and proper pronouns are added, resulting in the following query: "ما مرض يصيب البنكرياس وهو يعالج بالأنسولين؟".

VI. LIMITATIONS

To our knowledge, this work is the first step towards verbalizing SPARQL queries in Arabic. The main focus at this stage was to explore how the combination of morpho-syntactic features and dependency rules can be used for transforming SPARQL queries into legible Arabic sentences. However, Arabic is a rich language with a complex morphology when compared to English. Hence, there are several language and semantic issues that are not currently supported.

First, the processing of FILTER and GROUP BY constructs are not currently supported. The interpretation of these constructs is not straightforward due to the variety of formats they can have. In addition, they may entail significant modifications which can make the translation process complicated. Second, the generated Arabic sentence may require further realization by, for example, using appropriate pronouns, suffixes and prefixes. Arabic language is a free word order language; the sentence may have different forms of word order. Sometimes, it may be recommended to reorder words to improve the legibility of the sentence. However, sentence reordering has not been considered in this work.

VII. PRELIMINARY EVALUATION

A. Dataset

The main goal of this evaluation is to assess the ability of the proposed approach to verbalize SPARQL queries in Arabic. While there are plenty of OWL test data and questions in English [34], we are not aware of any ontology-based test data for Arabic that can be used for our evaluation. Therefore, we used the OWL dataset which we developed and used in previous works [2, 31]. This dataset consists of the Diseases ontology, of which an excerpt is shown in [2], and a set of 100 Arabic questions whose answers are in the ontology. This dataset was originally designed to assess natural language interfaces that take queries expressed in Arabic and generate SPARQL queries. As we aim to perform the opposite task, we recruited a human expert and explained the ontology content to him. We then asked the expert to choose twenty questions from the Arabic question set and convert them manually to SPARQL. The generated twenty SPARQL queries were then inputted to the system, and the output verbalizations were compared with the original Arabic queries. We also asked the expert to rate the similarity between each output query and the original query using a scale from 1 to 5, where 1 indicates a "weak similarity" and 5 indicates a "strong similarity".

B. Results and Discussion

Table III shows the twenty queries inputted to the system and their representations in SPARQL. The verbalizations generated by the system to these queries as well as the expert's ratings are also shown in the columns to the left. Overall, fifteen of the twenty queries (75%) were rated "4" or "5", two queries were rated "3", while three queries were rated "1" or "2". This result indicates that the system could verbalize most SPARQL queries in Arabic. On comparing the generated queries with the original ones, we found they were highly similar, expect few missing pronouns and word prefixes such as "الـ" that can make the output better readable. However, the expert indicated that all queries rated "4" and "5" conveyed the same meanings as the original ones, and that they were fairly legible.

Looking at the queries that got lower ratings, we found that the system failed to verbalize them for different reasons: For example, query #7 was not translated correctly because our approach cannot handle FILTER clauses. Some queries such as queries #14 and #16 did not give the intended meanings and needed better rephrasing. In fact, verbalizing these queries to capture the intended meanings may need deep reasoning that goes beyond the capabilities our approach. For example, the system is not currently able to handle queries starting with the word: "أذكر".

VIII. CONCLUSIONS AND FUTURE WORK

In this research, we proposed an approach to generate Arabic verbalizations for SPARQL queries. The translation from SPARQL to natural language goes through several stages: First, Arabic translations of terms in SPARQL are extracted, and variables are mapped to their corresponding types. Second, we use a set of language dependencies, which we adapted from

TABLE III. EVALUATION RESULTS: INPUT QUERIES, OUTPUT VERBALIZATIONS AND THE EXPERT'S RATINGS ARE SHOWN

	Original Query	SPARQL Query	Output Query	Rate
1	ما الأمراض التي تصيب القلب؟	SELECT ?x WHERE {:?x rdf:type :Disease . ?x :infects :Heart}	ما مرض يصيب القلب؟	5
2	ما سبب الإصابة بمرض الزهري؟	SELECT ?x WHERE{ :Syphilis :caused_by ?x}	ما سبب يسبب الزهري؟	3
3	ما الأمراض التي تصيب القلب أو تسبب تضخمه؟	SELECT ?x WHERE {:?x rdf:type :Disease . {?x :infects :Heart} UNION {?x :causes : :Cardiac_Hypertrophy}}	ما مرض يصيب القلب أو هو يسبب تضخمه؟	5
4	ما هي الامراض المعدية؟	SELECT ?x WHERE {?x rdf:type :Infectious_Disease}	ما مرض معدي؟	3
5	ما المرض الذي من أعراضه إلتهاب السحايا أو ضعف الإبصار؟	SELECT ?x WHERE {{?x :has_symptom :Meningitis} UNION {?x :has_symptom :Low_Vision}}	ما مرض له عرض إلتهاب السحايا أو ضعف الإبصار؟	4
6	ما المرض الذي يشخص بتحليل هرمون التستوستيرون؟	SELECT ?x WHERE {:Testosterone_Diagnosis :diagnoses ?x}	ما مرض يشخص بتحليل هرمون التستوستيرون؟	5
7	ما الأدوية التي لا تسبب ألم المعدة ؟	SELECT ?x WHERE {?x rdf:type :cure . FILTER NOT EXISTS {?x : causes :Stomach_Pain}}	ما دواء يسبب ألم للمعدة ؟	1
8	ما الأمراض التي يسببها التدخين؟	SELECT ?x WHERE {?x :caused_by :Smoking}	ما المرض يسبب بالتدخين؟	5
9	ما المرض الذي يصيب الأمعاء ومن أعراضه الاسهال والقيء ؟	SELECT ?x WHERE {?x :infects :small_intestine . ?x :has_symptom :vomiting . ?x :has_symptom :Diarrhea}	ما مرض يصيب الأمعاء الدقيقة و له عرض الإسهال والقيئ؟	4
10	هل التبول اللاإرادي من الأمراض النفسية؟	ASK{ :Enuresis rdf:type :Mental_Disease}	هل التبول اللاإرادي نوع من مرض نفسي؟	4
11	ما المرض الذي يصيب بنكرياس ويسبب عسر الهضم؟	SELECT ?x WHERE {{?x :infects:Pancrias . ?x :causes :Indigestion}	ما مرض يصيب البنكرياس ويسبب عسر الهضم؟	5
12	ما أعراض وعلاج الإنفلونزا؟	SELECT ?x ?y WHERE { ?x :symptom_of :Influenza . ?y :cures :Influenza}	ما عرض الإنفلونزا و علاجها؟	4
13	ماذا يعالج دواء فلاجيل؟	SELECT ?x WHERE {?x :cured_by :Flagyl}	ما مرض يعالج بفلاجيل؟	5
14	أذكر بعض الأمراض النفسية؟	SELECT ?x WHERE {?x rdf:type :Mental_Disease}	ما مرض نوع من مرض نفسي؟	2
15	كيف يعالج الجدري؟	SELECT ?x WHERE {?x rdf:type :Cure . ?x :cures :Smallpox}	ما دواء يعالج الجدري؟	4
16	أذكر بعض أنواع جراحات القلب؟	SELECT ?x WHERE {?x rdf:type :Heart_Surgery}	ما جراحة نوع من جراحة القلب؟	2
17	ما المرض الذي يصيب البنكرياس ويسببه نقص الأنسولين؟	SELECT ?x WHERE { ?x :infects :Pancreas. ?x :caused_by :Lack_of_Insulin}	ما مرض يصيب البنكرياس ويسبب بنقص الأنسولين؟	4
18	ما الأدوية التي تسبب الإجهاض؟	SELECT ?x WHERE {?x rdf:type :Cure . ?x:causes :Abortion}	ما دواء يسبب الإجهاض؟	5
19	ما أعراض التوحد؟	SELECT ?x WHERE {:Autism :has_symptom ?x }	ما عرض التوحد؟	5
20	ما هي الامراض التي تصيب الدم و من أعرضها ارتفاع ضغط الدم وتسبب عدم انتظام ضربات القلب؟	SELECT ?x WHERE {?x :infects :Blood. ?x :causes :Arrhythmias . ?x :has_symptom :High_Blood_Pressure}	ما مرض يصيب الدم وله عرض إرتفاع ضغط الدم ويسبب عدم انتظام ضربات القلب؟	5

Stanford dependencies of English language, and a set of handcrafted rules to structure words into valid sentences. Third, the linguistic realization of the sentence is boosted by exploiting morpho-syntactic analysis and NLP techniques. Finally, aggregation and redundancy elimination is performed to improve the legibility of the sentence.

There are many directions to extend this work: The limitations of the approach will be explored, especially the processing of FILTER clauses. We will also explore the expansion of Arabic dependencies in order to support additional structures of Arabic sentence, and hence, support the verbalization of more complex SPARQL queries. The approach will be assessed with a larger query set and different ontologies. We will also explore how the approach can be integrated into ontology-based applications to benefit users in practice.

REFERENCES

[1] http://research.microsoft.com/en-us/projects/atks/

[2] I. Al Agha, "Using Linguistic Analysis to Translate Arabic Natural Language Queries to SPARQL", International Journal of Web & Semantic Technology, Vol. 6, No. 3, pp. 25-39, 2015

[3] S. Shekarpour, A. -C. Ngonga Ngomo, S. Auer, "Question answering on interlinked data", 22nd International Conference on World Wide Web, pp. 1145-1156, Rio de Janeiro, Brazil, May 13 - 17, 2013

[4] J. Perez, M. Arenas, C. Gutierrez, "Semantics and complexity of SPARQL", ACM Transactions on Database Systems, Vol. 34, No. 3, Article No. 16, pp. 1-45, 2009

[5] H. Piccinini, M, A. Casanova, A. L. Furtado, B. P. Nunes, "Verbalization of rdf triples with applications", ISWC-Outrageous Ideas track, 2011

[6] M. Beseiso, A. R. Ahmad, R. Ismail, "A Survey of Arabic language Support in Semantic web", International Journal of Computer Applications, Vol. 9, No. 1, pp. 35-40, 2010

[7] A. -C. Ngonga, L. Buhmann, C. Unger, J. Lehmann, D. Gerber, "Sorry, i don't speak SPARQL: translating SPARQL queries into natural language", 22nd international conference on World Wide Web. International World Wide Web Conferences Steering Committee: Rio de Janeiro, Brazil. pp. 977-988, 2013

[8] B. Ell, D. Vrandečić, E. Simperl, "Spartiqulation: Verbalizing sparql queries", Lecture Notes in Computer Science, Vol. 7540, pp. 117-131 2015

[9] E. Kaufmann, A. Bernstein, "How useful are natural language interfaces to the semantic web for casual end-users?", 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, pp. 281-294, November 11-15, 2007

[10] E. Kaufmann, A. Bernstein, R. Zumstein, "Querix: A natural language interface to query ontologies based on clarification dialogs", 5th International Semantic Web Conference (ISWC 2006), pp. 980-981, 2006

[11] C. Pradel, O. Haemmerlé, N. Hernandez, "Natural language query interpretation into SPARQL using patterns", 4th International Workshop on Consuming Linked Data-COLD 2013, pp. 1-12, 2013

[12] S. Ferré, "SQUALL: The expressiveness of SPARQL 1.1 made available as a controlled natural language", Data & Knowledge Engineering.Vol. 94, No. 1, pp. 163-188, 2014

[13] G. Aguado de Cea, A. Bañón, J. Bateman, M. S. Bernardos, M. Fernández-López, A. Gómez-Pérez, E. Nieto, A. Olalla, R. Plaza, A. Sánchez, "ONTOGENERATION: Reusing domain and linguistic ontologies for Spanish text generation", Workshop on Applications of Ontologies and Problem-Solving Methods European Conference on Artificial Intelligence (ECAI'98), Brighton, United Kingdom, August 1998

[14] D. Hewlett, A. Kalyanpur, V. Kolovski, C. Halaschek-Wiener, "Effective NL Paraphrasing of Ontologies on the Semantic Web", End User Semantic Web Interaction Workshop, CEUR-WS Proceedings, Vol. 172, 2011

[15] N. T. Dong, L. B. Holder, "Natural Language Generation from Graphs", International Journal of Semantic Computing.Vol. 8, No. 3, pp. 335-384, 2014

[16] G. Wilcock, "Talking owls: Towards an ontology verbalizer", Human Language Technology for the Semantic Web and Web Services, Vol. 3, No. 1, pp. 109-112, 2003

[17] A. Third, S. Williams, R. Power, "OWL to English: a tool for generating organised easily-navigated hypertexts from ontologies", 10th International Semantic Web Conference (ISWC 2011), 23 - 27 Oct 2011, Bonn, Germany.

[18] K. Kaljurand, N. E. Fuchs, "Verbalizing OWL in Attempto Controlled English", OWL: Experiences and Directions Workshop (OWLED), Third International Workshop, Austria, June 6-7, 2007

[19] G. Koutrika, A. Simitsis, Y. E. Ioannidis, "Explaining structured queries in natural language", IEEE 26th International Conference on Data Engineering, pp. 333-344, USA, March 1-6, 2010

[20] N. Bouayad-Agha, G. Casamayor, L. Wanner, "Natural language generation in the context of the semantic web", Semantic Web Journal (under review)

[21] D. Gerber, A.-C. Ngonga Ngomo, "Extracting multilingual natural-language patterns for rdf predicates", Knowledge Engineering and Knowledge Management, pp. 87-96, 2012

[22] B. Ell, A. Harth, "A language-independent method for the extraction of RDF verbalization templates", 8th International Natural Language Generation Conference, pp. 26, 2014

[23] W. Zheng, L. Zou, X. Lian, J. X. Yu, S. Song, D. Zhao, "How to Build Templates for RDF Question/Answering: An Uncertain Graph Similarity Join Approach", 2015 ACM SIGMOD International Conference on Management of Data, pp. 1809-1824, 2015

[24] N. E. Fuchs, "First-order reasoning for attempto controlled english", Controlled Natural Language, pp. 73-94, 2012

[25] J. Danaparamita, W. Gatterbauer, "QueryViz: helping users understand SQL queries and their patterns", 14th International Conference on Extending Database Technology, pp. 558-561, 2011

[26] A. Kokkalis, P. Vagenas, A. Zervakis, A. Simitsis, G. Koutrika, Y. Ioannidis, "Logos: a system for translating queries into narratives", 2012 ACM SIGMOD International Conference on Management of Data, USA, pp. 673-676, 2012

[27] L. Al-Safadi, M. Al-Badrani, M. Al-Junidey, "Developing ontology for Arabic blogs retrieval", International Journal of Computer Applications.Vol. 19, No. 4, pp. 40-45, 2011

[28] F. Z. Belkredim, F. Meziane, "DEAR-ONTO: a derivational Arabic ontology based on verbs", International Journal of Computer Processing of Languages, Vol. 21, No. 3, pp. 279-291, 2008

[29] N. Soudani, I. Bounhas, B. El Ayeb, Y. Slimani, "Toward an Arabic Ontology for Arabic Word Sense Disambiguation Based on Normalized Dictionaries", On the Move to Meaningful Internet Systems: OTM 2014 Workshops, pp. 655-658, Confederated International Workshops: OTM Academy, OTM Industry Case Studies Program, C&TC, EI2N, INBAST, ISDE, META4eS, MSC and OnToContent 2014, Amantea, Italy, October 27-31, 2014

[30] A. Y. Mahgoub, M. A. Rashwan, H. Raafat , M. A. Zahran, M. B. Fayek, "Semantic Query Expansion for Arabic Information Retrieval", Arabic Natural Language Processing Workshop, Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, ACL, 2014

[31] I. Al Agha, A. Abu-Taha, "AR2SPARQL: An Arabic Natural Language Interface for the Semantic Web", International Journal of Computer Applications.Vol. 125, No. 6, pp. 2015

[32] I. Al Agha. "Diseases Ontology", available at: https://code.google.com/p/ar2sparql/

[33] Stanford Types Dependencies Manual, available at: http://nlp.stanford.edu/software/dependencies_manual.pdf

[34] Mooney Natural Language Learning Data, available at: https://files.ifi.uzh.ch/ddis/oldweb/ddis/research/talking-to-the-semantic-web/owl-test-data/