An Ontology-driven DBpedia Quality Enhancement to Support Entity Annotation for Arabic Text

Adham Kahlawi
Department of Statistics, Computer Science Applications, University of Florence, Florence, Italy

***Abstract*—Improving NLP outputs by extracting structured data from unstructured data is crucial, and several tools are available for the English language to achieve this objective. However, little attention has been paid to the Arabic language. This research aims to address this issue by enhancing the quality of DBpedia data. One limitation of DBpedia is that each resource can belong to multiple types and may not represent the intended concept. Additionally, some resources may be assigned incorrect types. To overcome these limitations, this study proposes creating a new ontology to represent Arabic data using the DBpedia ontology, followed by an algorithm to verify type assignments using the resource's title metadata and similarity between resources' descriptions. Finally, the research builds an entity annotation tool for Arabic using the verified dataset.**

Keywords—Entity annotation; semantics annotation; DBpedia; Arabic language; ontology; semantic web; linked open data

I. Introduction

The volume of unstructured text data increases daily, which increases the need for methodologies to help understand and classify this information. Since such data is not structured, the best way to understand it is by adding a metadata to it; consequently, it can be converted to semi-structured data. Part of Speech (POS) tagging, morphological annotation, structural annotation, pragmatic annotation, syntactic annotation, semantic annotation, pragmatic annotation, and stylistic annotation are several methodologies used in Natural Language Processing (NLP) to improve the results of the text analyze. One of the biggest sources of text data that is edited collaboratively from different users is Wikipedia [1]; furthermore, it covers a large number of topics, and it is a free information source. As a consequence, a specific type of semantic annotation has been involved based on Wikipedia called wikification [2]. Thus, Wikipedia has become training data for multiple models and not just a source of information. DBpedia is a project to convert Wikipedia data into structured data and make this data not only available on the web but also machine-readable [3]. DBpedia is available in different languages with cross-language being available to identify the same concept. In other words, the semantic structure of DBpedia allows the connection between more than one label written in more than one language with the same concept. One of these languages is Arabic, which makes DBpedia a good choice to be the information source to build entity annotation for Arabic text. Nonetheless, the transformation process that DBpedia underwent caused some problems, such as the use of the Wikipedia category system does not form a complete topical classification because there exist cycles between the categories, sometimes representing a loose connection between the Wikipedia articles [4]. Consequently, DBpedia data is not high quality. These problems can be summarized into two main categories. The first category, the types to which the DBpedia resources[1] belong to were extrapolated from the DBpedia ontology, but this ontology contains a large number of classes and subclasses. Thus, the single resource can belong to more than one type; meanwhile, these types do not necessarily refer to the same concept. In the second category, the process of assigning types to resources was not accurate enough; therefore, can find some resources which were assigned a type that does not correspond to their concept, or some resources which were assigned more than one type but at least one of these types does not correspond to their concept.

To build an efficient tool, the quality of the data taken from DBpedia has to be improved by building a methodology from several steps. The methodology begins from building a new ontology inspired by the DBpedia ontology, up to validating the information through the content of the text itself or by using the measurement of similarity between texts.

The rest of this paper is structured as follows; Section II discusses the related work. Section III shows what DBpedia is and how it is built. Section IV explains how to obtain data from DBpedia. Section V describes the data collected. Section VI explains the methodology that was followed in this article. Section VII evaluates the methodology. Finally, Section VIII concludes the paper.

II. Related Work

The ability to give metadata to text data has been established as a strategic task for understanding and analyzing texts. Cucerzan [5] seeks through his large-scale system for the recognition and semantic disambiguation to name entity by using data extracted from Wikipedia. Through a link between a single information presented in Wikipedia with the document in which presented and with the category tags of this document. This system passes through three stages. The first stage is to surface from entity mappings, which use the information of the titles of entity pages, the titles of redirecting pages, the disambiguation pages, and the references to entity pages in other Wikipedia articles. The second stage is category information, which takes advantage of the classifications built by Wikipedia contributors. The third stage is contexts, which use the information present in the entity page and other pages

[1] A resource is a basic object encapsulating a "real" thing like a text, image or whatever. This object is accessible within the server according to its Uniform Resource Identifier (URI) because each URI is unique at the global level. All the information that describes the resource itself is stored using the Resource Description Framework (RDF).

that explicitly refer to the same entity. Mihalcea and Csomai [6] integrate two algorithms into one system called Wikify, where the first algorithm identifies and extract the important words in the input text, and the second algorithm assigns each word of the output of the first algorithm with the correct page of Wikipedia. According to Milne and Witten [2], the Wikify system will make mistakes because of its dependence on a probabilistic value of word link that is calculated by the number of a link of the same word in different articles on Wikipedia with a specific article. For these reasons, they developed a new system that takes into consideration not only the word but also the context surrounding this word.

On the other hand, Kulkarni and others [7] proposed a new system that is not directly human interpretable, but downstream indexing, search, and mining. This system is an optimization of the previous one through investigating practical solutions based on local hill-climbing, rounding integer linear programs, and pre-clustering entities followed by local optimization within clusters. TAGME is a software system proposed by Ferragina and Scailla [8] which addresses the problem of cross-referencing text fragments with Wikipedia pages. In particular, TAGME took advantage of previous work and assigned it to small texts such as tweets. Makris et al. [9] proposed techniques that enhanced the TAGME to be applied to different approaches for Wikipedia disambiguation. These techniques improve the quality of Wikipedia by auxiliary information provided by more formal knowledge resources like WordNet [10], which is a large lexical network where concepts/senses are represented by so called synsets. In addition, these techniques employed the PageRank of the Wikipedia pages as an extra factor for disambiguation. Wikifier is a semantic annotation approach presented by Brank and others [11], which supports different languages that are available in Wikipedia; furthermore, it is suitable for parallel processing and supports various minor heuristics. Wikifier refinements is an effort to improve the performance of other approaches; indeed, it uses Wikipedia class membership to ignore certain types of concepts and use the indicator of word frequency to ignore the common words. Consequently, Wikifier reduces the noise in the annotation output. WEXEA, a Wikipedia EXhaustive Entity Annotation system [12] proposed by Strobl and others. the approach aims to create an annotated text corpus include all mentions in Wikipedia instead of simply depend on already existing links between Wikipedia pages; as a consequence, that can be more useful in downstream tasks and can introduce unnecessary errors. There have been several attempts to build tools for the Arabic language. In the following, we will list the most important of these attempts. In 2008 Benajiba et al. [13] used contextual, lexical, part-of-speech, and other features to train a Named Entity Recognition model using an SVM-based approach. The model can recognize four (Person, Location, Organization, and Miscellaneous). Next, in 2012 Al-Jumaily et al. [14] built a real-time named entity recognition system using the data from DBPedia and other sources. The system can recognize three entities only (Person, Location and Organization); the system can also extract the linguistic roots (nouns and verbs). Then, in 2014 Yosef et al. [15] used the Wikipedia Arabic version to create a Named entity disambiguation framework. The framework structure contains four principal concepts entity repository, name entity dictionary, entity description and entity-entity relation. Meanwhile, the framework uses the connection between the Arabic version and the English one of Wikipedia to control the data quality. To test the framework, the authors chose ten news Arabic articles and manually annotated them. After that, in 2015, Al-Yahya et al. [16] presented a lexical semantic annotation based on an ontology that contains six classes that build a three-level hierarchical structure. The higher level is Linguistic Concepts, and the lower level is Words. Consequently, in 2016, Al-Qawasmeh et al. [17] created a similar tool based on DBPedia, where they made an ontology consisting of three classes and filled it in the DBPedia's individuals that belong to these three classes. The authors use a similar algorism to optimize the performance where the algorism finds the nearest entity from the dataset to the input entity. Finally, in 2018, Albukhitan et al. [18] built a tool using deep learning to recognise three entities (Food, Nutrition and Health). The tool will assign the closest ontological class to the input text as a named entity using the weighted candidate vectors and word2vec model. Jarrar et al. [29] introduce Wojood, a corpus for nested Named Entity Recognition in Arabic, consisting of 550K tokens manually annotated with 21 entity types, including nested entities, with a strong inter-annotator agreement. The corpus was used to train a nested NER model achieving a micro F1-score of 0.884, and all resources are publicly available. Al-Thubaity et al. [30] present the COVID-19 Arabic Named Entities Recognition (CAraNER) dataset, consisting of 55,389 tokens from Saudi Arabian newspaper articles labelled with five named-entity tags. The paper also evaluates the dataset using four BERT-based Arabic language models, with AraBERTv0.2-large achieving the highest F1 macro measure of 0.86.

III. Baseline: Dbpedia

Wikipedia is one of the large knowledge sources, and is created by the contribution of thousands of users through writing the articles using natural languages. On the other hand, Wikipedia contains info box, images, geo-coordinates, and categorization information which can be considered as structured information. Moreover, this structure information has been presented by different languages [19]. The structured information of Wikipedia was the basis for building the DBpedia through an open-source extraction framework; as a consequence, DBpedia has been considered as a multilingual and multidomain knowledge base. Each concept of DBpedia is described by a corresponding Wikipedia page and is identified by a Uniform Resource Identifier (URI) [20]. The use of URI and the Resource Description Framework (RDF) has made the DBpedia relatively stable, machines readable, and commonly used ontology. The DBpedia ontology can be used in the integration of different languages and the organization of extracted data [21]. Lehmann and others [22] illustrated the process of data extraction from Wikipedia through Fig. 1. The data extraction process passes through several stages that start from the inputs which are Wikipedia pages; secondly, it moves to the parsing stage that transforms the pages into an Abstract Syntax Tree; thirdly, the extraction stage which converts different parts of pages to RDF by following the next steps:

- the creation of DBpedia ontology terms like the data property through the mapping of infobox structure manually;
- mapping the information in the infobox to RDF taking into consideration the DBpedia ontology;
- extracting a single feature from pages such as a label or geographic coordinates;
- extracting aggregated data from all pages like word counts as further description.

Furthermore, to improve the DBpedia by adding the Arabic language, Ismail et al. [23][23] [23] made special data extraction for Arabic Wikipedia pages through the mapping of infoboxes to the DBpedia ontology by different mapping extractors.

The data extraction based on infoboxes was of great importance in the automation of extraction; on the other hand, it causes some quality issues. Consequently, several papers have been published discussing these issues. For instance, the quality problems resulting from incorrect or missing information like incorrect values, data types, and links were discussed by Zaveri et al. [24], several automatic quality tests were provided by Kontostas et al. [25] to be applied to Linked Open Data (LOD) dataset of DBpedia, the DBpedia accessibility quality was tested by a Linked Data Quality Model that was developed by Radulović et al. [26], and the use of machine learning was proposed by Rico et al. [27] for the detection of incorrect mappings. Lakshen et al. [28] focused on identifying the quality of Arabic DBpedia since it contains problems different from the problems of other languages like the presentation of characters as symbols, the use of Hindu numerals that can create wrong values in numerical data, and occurrence of different names for the same attribute.

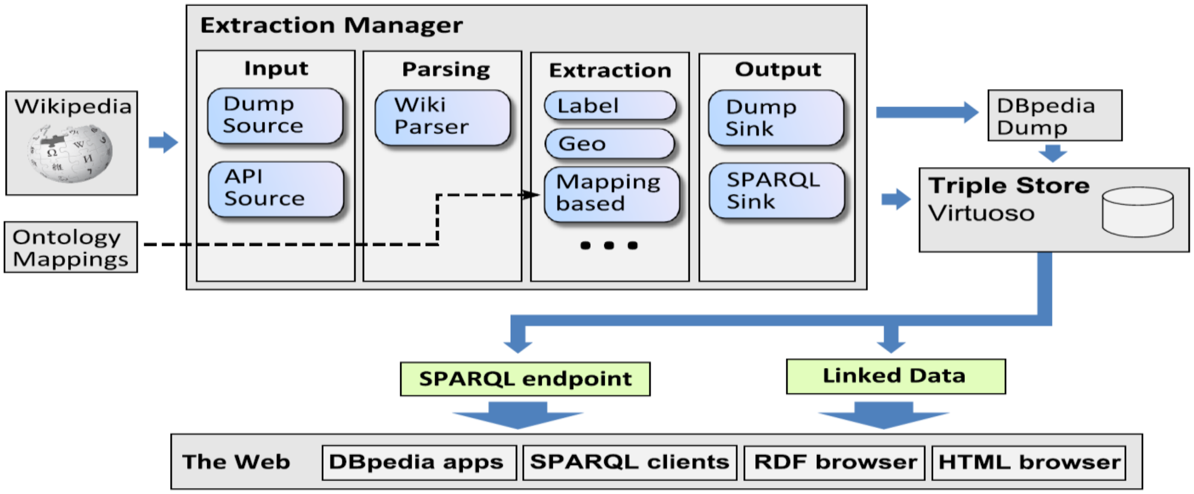

Fig. 1. Overview of DBpedia extraction framework [15].

IV. Data Collection

The latest releases of core data from en.wikipedia.org have been published by the DBpedia protect on the first of July 2020; also the collection URI is https://databus.dbpedia.org/dbpedia/collections/latest-core. However, the collection does not contain downloadable files for the Arabic language; for this reason, the data has been acquired by applying SPARQL Query at a DBpedia public SPARQL endpoint “http://dbpedia.org/sparql” using python function.

```
        def apply_query(query):
            sparql=
SPARQLWrapper("http://dbpedia.org/sparql")
            sparql.setQuery(query)
            sparql.setReturnFormat(JSON)
            results = sparql.query().convert()
            df=
pd.json_normalize(results["results"]["bindings"])
            return df
```

public SPARQL endpoint has a limit as the following[31] :
“ *ResultSetMaxRows = 10000*
MaxQueryExecutionTime = 120 (seconds)
MaxQueryCostEstimationTime = 1500 (seconds)
Connection limi = 50 (parallel connections per IP address)
maximum request rate = 100 (requests per second per IP address, with an initial burst of 120 requests)”

As a consequence, the data was collected following the steps below:

a) identify the types of all the instances.

The DBpedia collection offers a special file that contains the type of all instances.

b) building the DBpedia Arabic dataset.

At this point, a dataset containing instances URI, label, description, and type will be obtained by sending a SPARQL query to DBpedia SPARQL endpoint through a specific Python code. The SPARQL query has been built and has been sent for each type obtained in the previous step.

where_text = """ ?uri rdfs:label ?label;

```
                    rdf:type   ?type.
               optional{}
              filter(lang(?label)="ar")
            filter(?type in(<{}>))
    """
optional_text = """{
              ?uri rdfs:comment ?descrption.
             filter(lang(?descrption)="ar")
            }"""
for i in range(0,len(type_list)):
   arabic_dataset_query = f"""PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
   PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
   SELECT distinct ?uri ?label  ?descrption ?type WHERE {'{'}
   {where_text.format(optional_text,type_list[i])}
   {'}'}
   """
   arabic_label_df=
arabic_label_df.append(apply_query(arabic_datset_query))
```

V. Data Description

In this chapter, we will discuss the validity of the dataset from two different points of view. Firstly, the dataset description and the affiliations of each resource; secondly, the formal and grammatical structure of the resources' label.

A. Description of the Dataset obtained from DBpedia

As shown in Fig. 2, the number of URI of the resources without repetition within the dataset is greater than the number of resources label without repetition. As a consequence, this information indicates the problem of the existence of a group of resources within the dataset that have the same name but have more than one URI.

On the other hand, the total data in the dataset is greater than the number of resources label without repetition. For this reason, there is a problem of the presence of some resources that belong to more than one type; in fact, Fig 3 shows that resources may belong to more than one type at the same time and the max is seven types for example: this URI http://dbpedia.org/page/Pomegranate_soup refer to Pomegranate soup resource that has two different types, the first one is dbo:Food but the second one is dbo:GivenName that usually use to indicate a person name. However, it is not necessary that these different types refer to different concepts; nonetheless, they may refer to the same concept with different names or refer to the sub-concepts of the same concept.

B. The Formal and Grammatical Structure

Arabic labels contain additional information placed in brackets to clarify the meaning of the label; on the other hand, this method of adding additional information is not considered as one of the common methods used in the Arabic language to attend this goal. For example, the URI http://dbpedia.org/resource/Pato has the Arabic label written in this way "باتو(رياضة)" which means Pato (sport); however, the English label is written with only the word Pato. The dataset contains 63747 Arabic labels contain additional information placed in brackets.

As for the linguistic rules, the rules of definite or indefinite articles in the Arabic language is different from other languages like English. The English language uses the articles that are placed before the noun, indicating whether the word is definite or indefinite. In contrast, the Arabic language adds two letters to the word at the beginning to differentiate between definite or indefinite nouns; consequently, the natural language processing will consider the definite word different from the same word in the indefinite form.

For instance, the URI that has Cartilage as English label, this label can be written inside any English text as "a cartilage" or "the cartilage"; in contrast, the Arabic label of this URI is غضروف that can be written inside any Arabic text as "غضروف" or "الغضروف".

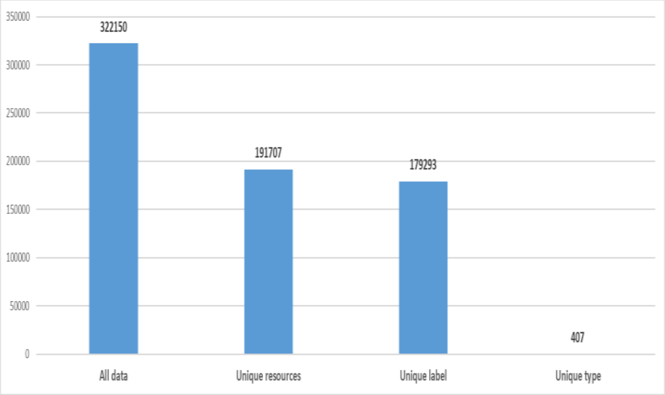

Fig. 2. Dataset description.

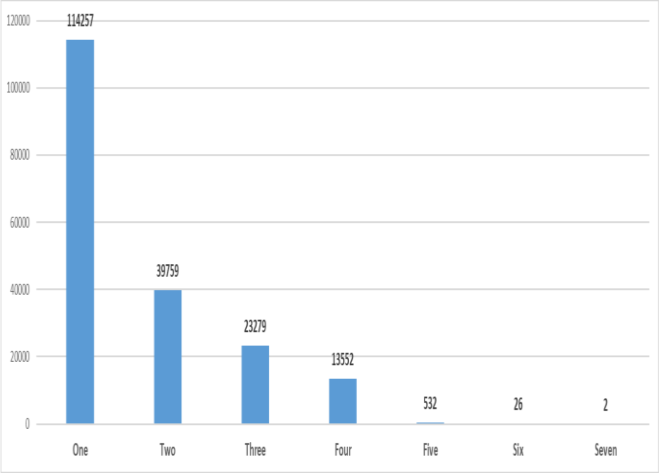

Fig. 3. Frequency of resources depending on the number of types.

VI. Methodology

The methodology seeks to reduce the number of types that some resources can belong to; in addition, seeks to verify the validity of the types that have been assigned to the resources. Finally, the data obtained from achieving the two previous goals will be used to build a tool to annotate the Arabic text. Fig. 4 shows that the methodology depends on the implementation of a set of successive steps to achieve its objective, which are described as follows:

A. Develop a New Ontology to Improve Data Quality

Each resource in the data belongs to one or more types as shown in Fig. 3; indeed, these types belong to DBpedia ontology. By referring to DBpedia ontology, it can be seen that the group of types to which the resource belongs does not necessarily belong to different concepts. However, these types may belong to the same general concept; consequently, it represents one of the oldest data quality problems faced by machine learning technologies [32]. Therefore, this study proposes to create a new ontology inspired by the DBpedia ontology and correspond to the types of Arabic dataset in order to improve the quality of data and reduce the number of types to which belongs every resource [33][34]. Afterward, the basic types to which the data belong were transformed into individuals belonging to the classes of new ontology that represent the general concept of it. We must point out that building the new ontology depends on human experience; therefore, it will be a manual process, while the process of individual transformation for it will be automatic. Fig. 5 represents the DBpedia Arabic resources ontology, while Table I represents its metrics.

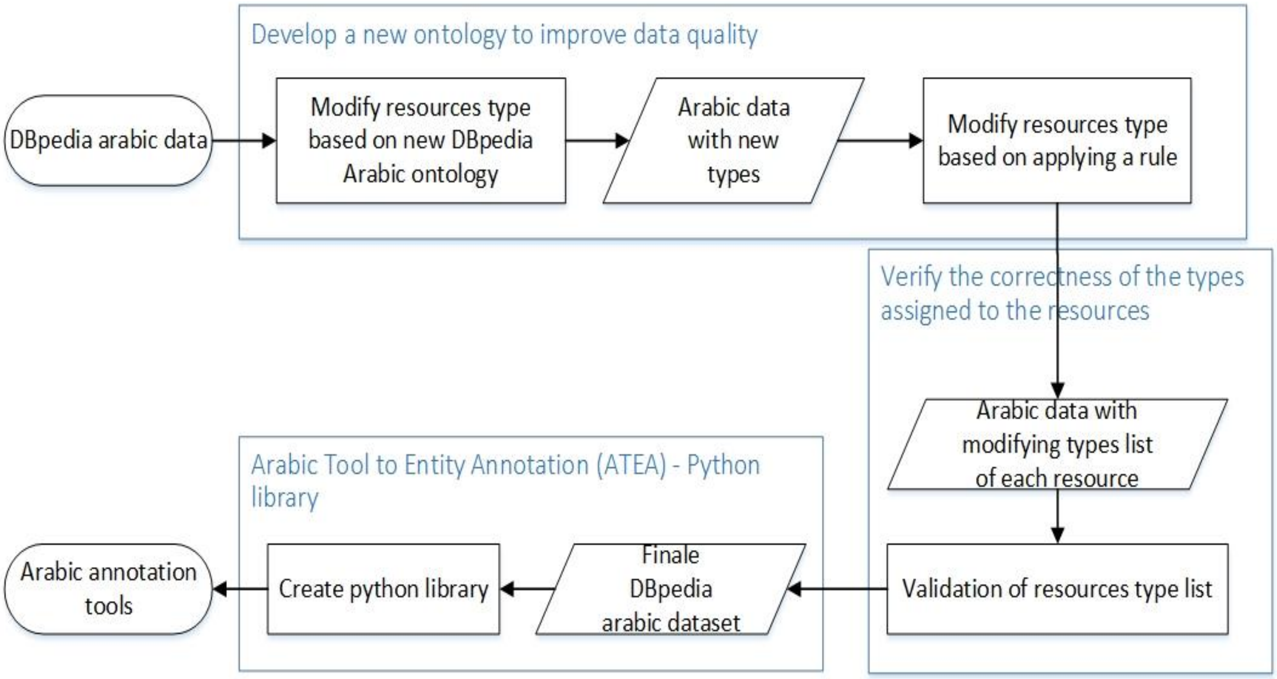
Develop a new ontology to improve data quality
DBpedia arabic data
Modify resources type based on new DBpedia Arabic ontology
Arabic data with new types
Modify resources type based on applying a rule
Verify the correctness of the types assigned to the resources
Arabic data with modifying types list of each resource
Validation of resources type list
Arabic Tool to Entity Annotation (ATEA) - Python library
Finale DBpedia arabic dataset
Create python library
Arabic annotation tools

Fig. 4. The process underlying our methodology.

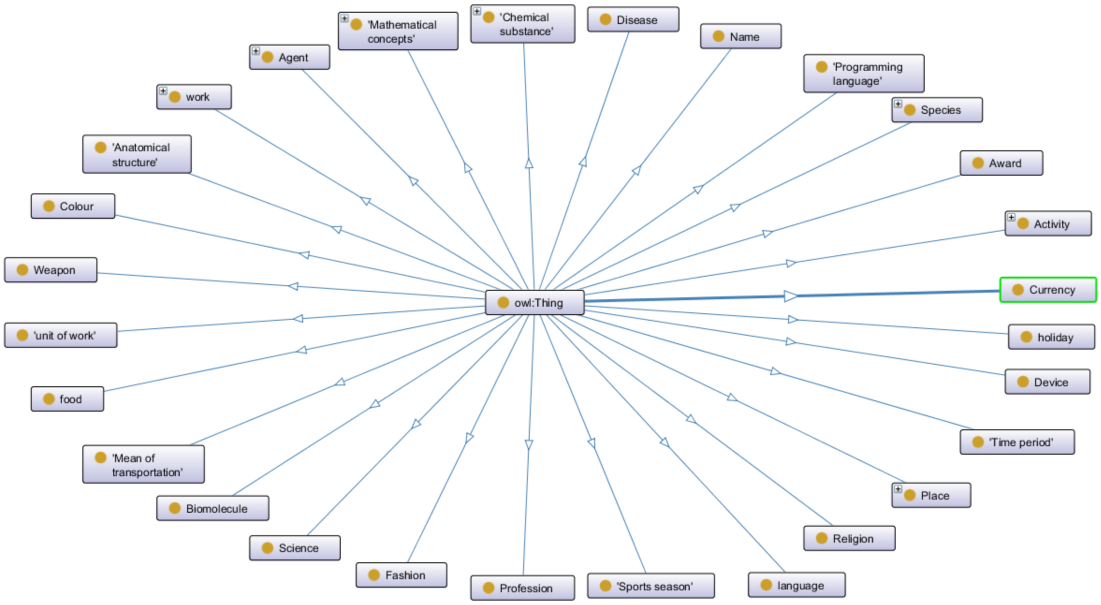

Fig. 5. DBpedia Arabic resources ontology.

TABLE I. DBPEDIA ARABIC RESOURCES ONTOLOGY METRICS

Ontology metrics	description
Metrics	
Axiom	240
Logical axiom count	106
Declaration axioms count	134
Class count	134
Class axioms	
SubClassOf	**106**

As a consequence of applying the previous mapping process to the data, each element belongs to a new type or group of types. Based on the new ontology, the following rule was applied to reduce the number of the types to which each source belongs.

Rule:

If there are two types one of which is a subclass of the other in the group of types to which a resource belongs; consequently, the type that represents a subclass is retained and the other is deleted from the group of types to which this resource belongs.

B. *Verify the Correctness of the Types Assigned to the Resources*

The DBpedia data is considered as structure data that was built through a process of mapping from Wikipedia; nonetheless, this process caused some errors in assigning the correct type to some resources. For this reason, this study proposed an algorithm to verify the validity of the type assigned to each resource and correct it in the event that an error is discovered. This algorithm is shown in Fig. 6 based on thirteen actions that are described as follows:

- Action 1 - Create URI_Type dictionary: A subset will be extracted from the dataset which contains the URI of each resource and its type or their types.

- Action 2 - Create URI_label dictionary: A subset will be extracted from the dataset which contains the URI of each resource and its Arabic label.

- Action 3 - Extract words in parentheses: The words written between parentheses in some labels of resource will be extracted.

- Action 4 - Associate words with the type they refer to: The list of words that result from the previous action will be checked manually if it refers to a specific type; then, a dataset of words and the types that refer to will be built.

- Action 5 - Assign to the resource URI the extracted word from parentheses presented in its label: Will build a dataset of URI of each resource and the extracted word written between parentheses in the label of resources, or the value of Null if the label does not include words between parentheses.

- Action 6 - Assign the word's type to the URI: The sources whose label contains an explanatory word and this word refer to a specific type. In this action, this type will be assigned to the resource URI.

- Action 7 - Create URI_Description dictionary: A subset will be extracted from the dataset which contains the URI of each resource and its English description.

- Action 8 - Calculate the similarity between the descriptions: The similarity will be calculated using the Latent Semantic Indexing (LSI) model. LSI is an indexing and retrieval technique. LSI is a text mining that is able to calculate the similarity between the text data by projecting it into space with latent semantic dimensions. In other words, the similarity is calculated by the co-occurrence of each word and every single word in the documents. The measure ranges from 0 to 1 (the greater the more similar)[35, 36]. LSI model was trained by using the English description of each resource to calculate the percentage of similarity between the resources, and then this dataset was constructed by linking each resource with the best five resources similar to it.

- Action 9 - Get for each URI the type of the best five similar URI: The type of five most similar resources of each resource will be gotten; indeed, the type of similar resources will be gotten from two sources. The first "verified URI_type dictionary" will be used if a similar resource is one of the resources whose type has been verified in the action 6. The second "URI_type dictionary" will be used if a similar resource is one of the resources that have not been verified its type yet.

- Action 10 - Get the type list for each URI: the type list of each resource will be gotten.

- Action 11 - Calculate the intersection between the two list: The intersection value between the two lists resulted in action 9 and action 10 will be calculated.

- Action 12 - Assign the intersection values as type to the URI: The intersection value will be assigned to the resource URI as a type for it.

- Action 13 - Human verification to the URI type: All resources that were the results of the calculated intersection in the action 11 are Null, its type will be verified manually. Manual verification faces two cases. The first case, the type that was assigned to the resource is correct, so no action will be taken. The second case, the type that was assigned to the resource is not appropriate for it, so the type will be changed to a type that corresponds to the nature of the resource.

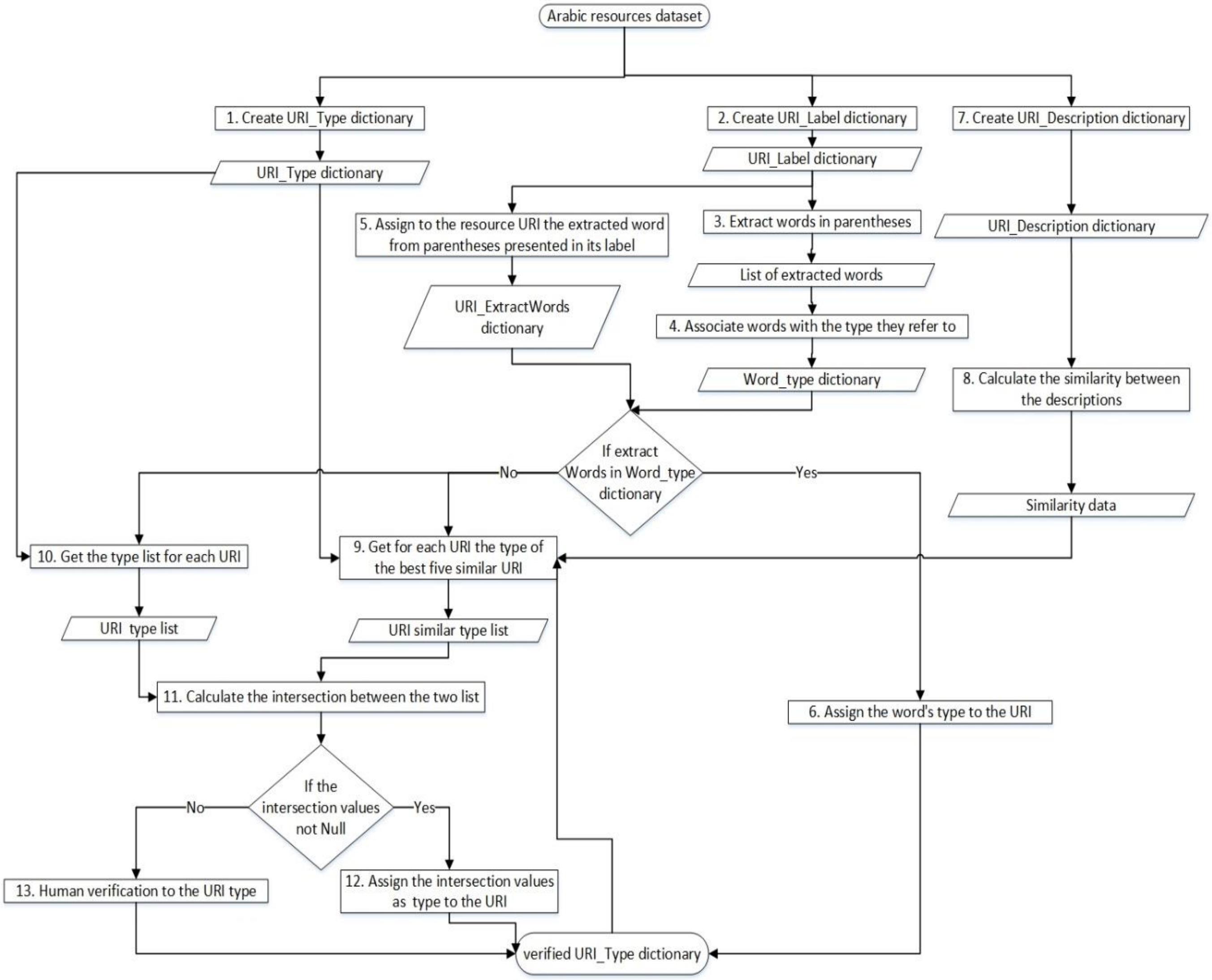
Arabic resources dataset
1. Create URI_Type dictionary
2. Create URI_Label dictionary
7. Create URI_Description dictionary
URI_Type dictionary
URI_Label dictionary
5. Assign to the resource URI the extracted word from parentheses presented in its label
3. Extract words in parentheses
URI_Description dictionary
List of extracted words
URI_ExtractWords dictionary
4. Associate words with the type they refer to
Word_type dictionary
8. Calculate the similarity between the descriptions
If extract Words in Word_type dictionary
No
Yes
Similarity data
10. Get the type list for each URI
9. Get for each URI the type of the best five similar URI
URI type list
URI similar type list
11. Calculate the intersection between the two list
6. Assign the word's type to the URI
If the intersection values not Null
No
Yes
13. Human verification to the URI type
12. Assign the intersection values as type to the URI
verified URI_Type dictionary

Fig. 6. An algorithm to verify the correctness of the types assigned to the resources.

C. Arabic Tool to Entity Annotation (ATEA) - Python Library

Fig. 7 illustrates the mechanism of ATEA library's work as it takes short text lake input; in contrast, its output contains different types of data. The first one represents the DBpedia Arabic resource ontology's types that belong to each entity detected in the text. The second type represents the DBpedia URI and DBpedia types that belong to each entity detected in the text. The last type represents the interpretation of the label of each entity detected in the text to different languages available in DBpedia.

ATEA includes four data set:

- URI_Label dataset: This dataset contains the entire Arabic resources label after applying two adjustments. First, all additional information attached to the primary label has been deleted; second, all the labels have been stemmed.
- Verified URI_Type dataset: This dataset contains all the resources types; indeed, these types follow the DBpedia Arabic resources ontology.
- URI_Type dataset: This dataset contains all the resources types obtained directly from DBpedia.
- This dataset contains the English label of all resources included in URI_Label dataset.

ATEA performed seven actions as following:

- Action 1 - Split text: This action is divided into two steps; first one is splitting the text based on the punctuation marks. The second step is tokenizing each subtext to all possible n gram word where n is between one and the number of words of the longest label in DBpedia Arabic resources.
- Action 2 - Find all entities: The intersection between the label list of DBpedia Arabic resources and the tokenization list will be calculated.
- Action 3 - Get the DBpedia Arabic resource ontology's type of each entity: Each element in the intersection list will be assigned to its type of the DBpedia Arabic resource ontology.

- Action 4 - Get the DBpedia URI and types of each entity: Each element in the intersection list will be assigned to its type and URI of the DBpedia.
- Action 5 - Interpret each entity to English language: Will be assigned to each element in the intersection list the English label of its represented resource in DBpedia.
- Action 6 - Integration of the three categories of data: The results of each item in the intersection list will be combined and organized into a dictionary that will represent the output of ATEA.

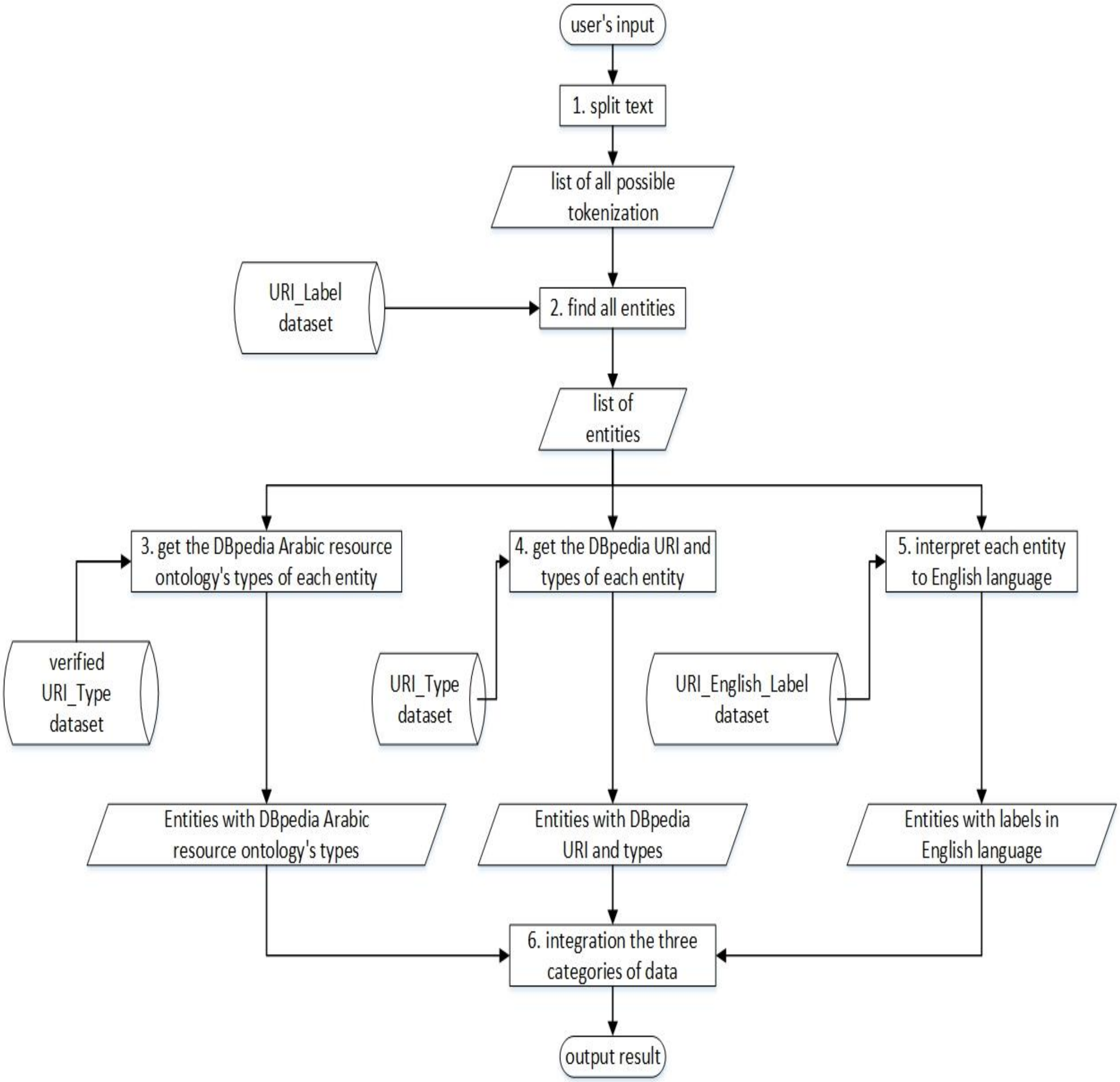
user's input
1. split text
list of all possible tokenization
URI_Label dataset
2. find all entities
list of entities
3. get the DBpedia Arabic resource ontology's types of each entity
4. get the DBpedia URI and types of each entity
5. interpret each entity to English language
verified URI_Type dataset
URI_Type dataset
URI_English_Label dataset
Entities with DBpedia Arabic resource ontology's types
Entities with DBpedia URI and types
Entities with labels in English language
6. integration the three categories of data
output result

Fig. 7. Arabic tool to entity annotation.

VII. Evaluation

Arabic resources collected from DBpedia had 407 different types and the resource may have up to seven types; consequently, each resource belongs to 1.68 types on average. The use of our developed DBpedia Arabic resources ontology reduced the number of types that the resources could belong to; as a result, DBpedia Arabic resources ontology grouped the types that belong to one concept together; thus, the number of types was reduced from 407 to 106. Indeed, Fig. 8 shows that the max number of types that any resource could belong to is five, and each resource belongs to 1.33 types on average. On the other hand, the use of DBpedia Arabic resources ontology left 739 resources without type, because these resources were related to the "Thing" type such as the resource of Whale that has the URI "http://dbpedia.org/resource/Whale". In addition, the implementation of the rule resulted in the max number of types that any resource could belong to reduce to three and each resource belongs to 1.08 types on average. At the end of the first step of the methodology, it is noted the optimization in the number of types that one resource can belong to; in fact, about 92% of the resources belong to one type.

In the second step of the methodology, firstly, 1688 words indicating specific types were extracted from information contained in parentheses in Arabic labels. Examples of these words are Novel, Book, Journal, Writer, Poet, Rocket, Tank, TV series, wrestling, Scientist, king, Restaurant, Church, programming language, Plant and bird. These words were used to verify the validity of the types to which the resources belong; indeed, if the Arabic label of any resource contains a word indicating a specific type and this type does not correspond to the resource type, the source type is modified to the type to which the word refers, as shown in Table II.

Secondly, the similarity between the descriptions of the resources was used. If the resource type corresponds to at least one of the five types of sources that are most similar to it; consequently, the resource type is considered correct. Otherwise, it will be manually verified. Table III shows some resources that have been verified manually.

As a result of the second step of the methodology, the max number of types that any resource could belong to is three types, but each resource belongs to 1.01 types on average. Moreover, about 99% of the resources belong to one type.

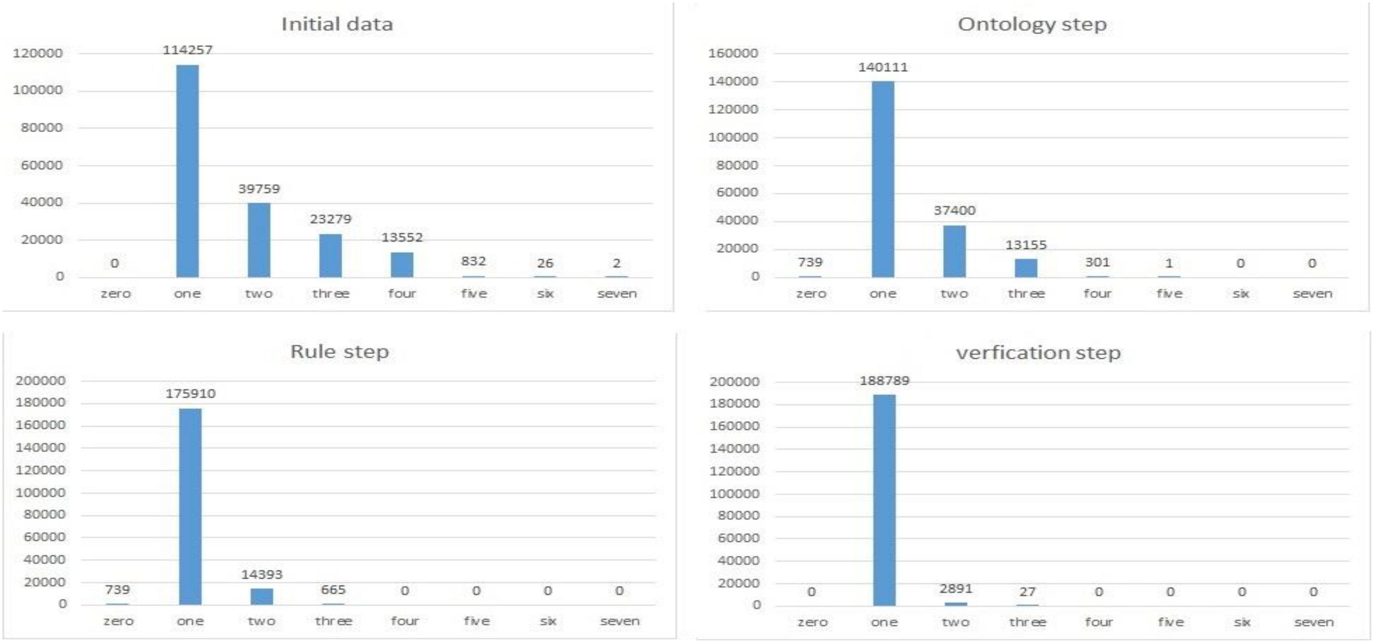

Fig. 8. Frequency of resources depending on the number of types that belong to for each methodology's step.

TABLE II. Examples of Text Verification Results

Resource URI	Arabic label	English label	DBpedia type	verify type
http://dbpedia.org/resource/Anthony_Wong_(Hong_Kong_actor)	أنتوني وونغ (ممثل هونغ كونغي)	Anthony Wong (Hong Kong actor)	Artist, Film	Artist
http://dbpedia.org/resource/Antalya_Province	أنطاليا (محافظة)	Antalya Province	PopulatedPlace, Athlete	PopulatedPlace
http://dbpedia.org/resource/Bell_AH-1Z_Viper	فايبر (مروحية)	Bell AH-1Z Viper	MeanOfTransportation	Weapon
http://dbpedia.org/resource/George_Hamilton_(actor)	جورج هاميلتون (ممثل)	George Hamilton (actor)	Film	Artist
http://dbpedia.org/resource/Ballade_(classical_music)	بالاد (موسيقى كلاسيكية)	Ballade (classical music)	Thing	MusicalWork

TABLE III. EXAMPLES OF HUMAN VERIFICATION RESULTS

Resource URI	**Arabic label**	**English label**	**DBpedia type**	**verify type**
http://dbpedia.org/resource /Orient_News	أورينت نيوز	Orient News	Band, Company	Media
http://dbpedia.org/resource/König-class_battleship	بارجة فئة كونيغ	König-class battleship	Band, MeanOfTransportation	Weapon
http://dbpedia.org/resource/ Sovereign_Military_Order_of_Malta	فرسان مالطة	Sovereign Military Order of Malta	EducationalInstitution, PopulatedPlace	MilitaryUnit
http://dbpedia.org/resource/Gezer	تل الجزر	Gezer	Event, MilitaryConflict	HistoricPlace
http://dbpedia.org/resource/ Burgundians	برغنديون	Burgundians	Language, Name	EthnicGroup
http://dbpedia.org/resource/Sabines	سابينيون	سابينيون	Language, Animal	EthnicGroup
http://dbpedia.org/resource/ Amr_Sobhy	عمرو صبحي	Amr Sobhy	Award, Writer	Writer
http://dbpedia.org/resource/Whale	حوت	Whale	Thing	Animal
http://dbpedia.org/resource/Narcotic	ناركوتي	Narcotic	Thing	Drug
http://dbpedia.org/resource/ Tank	دبابة	Tank	Thing	Weapon
http://dbpedia.org/resource/ Beef	لحم بقر	Beef	Thing	Food
http://dbpedia.org/resource/ Operetta	أوبريت	Operetta	Thing	Play
http://dbpedia.org/resource/ Train	قطار	Train	Thing	MeanOfTransportation

In the third step of the methodology, ATEA is built which takes short text as its input and then performs the following steps:

Firstly, the length of the text is measured, if the length of the text is equal or greater than 17 words, the text will be divided into n_ gram where n = 17; because of the max length Arabic label present in the data equal to 17. On the other hand, if the length of the text is less than 17 then the text will be divided into n_ gram where n equal to the length of the text.

Secondly, all n_gram obtained from the previous step are searched for; afterward, if any of them was found in the database. First of all, it will be annotated then it will be deleted from the text; last of all, if n equal to one, the finale results will be returned to the user; in contrast, if n greater than one, the previous steps will be repeated. This is the application of your method to an example. The example talks about Syria.

سورية دولة عربية يبلغ عدد سكانها ما يقارب 23 مليون منهم 93% عرب “ و 5% كرد و 2% أعراق أخرى مثل أرمن و تركمان. يعتنق السوريون أكثر من ديانة فنجد أن 90% إسلام و 7% مسيحية و 3% طوائف أخرى. يتكلم غالبية السكان لغة عربية. . تعاقبن على سوريا حضارات ممتالية ومن أشهرها إبلا وكانت هذه الحضارة العريقة والقوية قد ازدهرت في منتصف الألف الثالث قبل الميلاد و ” التي تم إكتشافها من قبل بعثة أثرية من جامعة روما سابينزا برئاسة باولو ماتييه.

“Syria is an Arab country with a population of approximately 23 million, of which 93% Arabs, 5% Kurds, and 2% other ethnicities such as Armenians and Turkmen. Syrians have more than one religion; indeed, we find that 90% Islam, 7% Christian, and 3% other sects. The majority of the population speaks Arabic. Successive civilizations took place in Syria, the most famous of which is Ebla. This ancient and powerful civilization had flourished in the middle of the third millennium BC, which was discovered by an archaeological mission from the University of Rome Sapienza headed by Paolo Mattei.”

The automatic entity annotation results are in JSON format as the following:

```
{
'جامعة روما سابينزا':
{'DBpedia_URI':
['http://dbpedia.org/resource/Sapienza_University_of_Rome'],
  'label': {'ar': ['جامعة روما سابينزا'],'en': ['Sapienza University of Rome']},
  'type': {'ar': ['موسسة اكاديمية'], 'en': ['Educational institution']},
  'DBpedia_type': ['http://dbpedia.org/ontology/University',
   'http://dbpedia.org/ontology/Organisation',
   'http://dbpedia.org/ontology/EducationalInstitution']},
 'لغة عربية':
{'DBpedia_URI': ['http://dbpedia.org/resource/Arabic'],
  'label': {'ar': ['لغة عربية'], 'en': ['Arabic']},
  'type': {'ar': ['لغة'], 'en': ['language']},
  'DBpedia_type': ['http://dbpedia.org/ontology/Language']},
 'إبلا':
{'DBpedia_URI': ['http://dbpedia.org/resource/Ebla'],
  'label': {'ar': ['إبلا'], 'en': ['Ebla']},
  'type': {'ar': ['مكان تاريخي'], 'en': ['Historic place']},
  'DBpedia_type': ['http://dbpedia.org/ontology/Country']},
  'سوريا':
{'DBpedia_URI': ['http://dbpedia.org/resource/Syria'],
  'label': {'ar': ['سوريا'], 'en': ['Syria']},
  'type': {'ar': ['مكان مأهول'], 'en': ['populated place']},
  'DBpedia_type': ['http://www.w3.org/2002/07/owl#Thing',
   'http://dbpedia.org/ontology/Place',
   'http://dbpedia.org/ontology/Country',
   'http://dbpedia.org/ontology/MusicalArtist',
   'http://dbpedia.org/ontology/PopulatedPlace']},
 'عرب':
{'DBpedia_URI': ['http://dbpedia.org/resource/Arabs'],
  'label': {'ar': ['عرب'], 'en': ['Arabs']},
  'type': {'ar': ['مجموعة عرقية'], 'en': ['Ethinc group']},
  'DBpedia_type': ['http://dbpedia.org/ontology/EthnicGroup',
   'http://dbpedia.org/ontology/Band']},
 'تركمان':
{'DBpedia_URI': ['http://dbpedia.org/resource/Turkmens'],
  'label': {'ar': ['تركمان'], 'en': ['Turkmens']},
  'type': {'ar': ['مجموعة عرقية'], 'en': ['Ethinc group']},
  'DBpedia_type':
['http://dbpedia.org/ontology/EthnicGroup']},
 'أرمن':
{'DBpedia_URI': ['http://dbpedia.org/resource/Armenians'],
  'label': {'ar': ['أرمن'], 'en': ['Armenians']},
  'type': {'ar': ['مجموعة عرقية'], 'en': ['Ethinc group']},
  'DBpedia_type': ['http://dbpedia.org/ontology/EthnicGroup',
   'http://dbpedia.org/ontology/Band']},
 'مليون':
{'DBpedia_URI': ['http://dbpedia.org/resource/Million'],
  'label': {'ar': ['مليون'], 'en': ['Million']},
  'type': {'ar': ['وحدة قياس'], 'en': ['Measure unit']},
  'DBpedia_type':
['http://dbpedia.org/ontology/Organisation']},
'مسيحية':
{'DBpedia_URI': ['http://dbpedia.org/resource/Christianity'],
  'label': {'ar': ['مسيحية'], 'en': ['Christianity']},
  'type': {'ar': ['دين'], 'en': ['Religion']},
  'DBpedia_type':
['http://dbpedia.org/ontology/EthnicGroup']},
 'إسلام':
{'DBpedia_URI': ['http://dbpedia.org/resource/Islam'],
  'label': {'ar': ['إسلام'], 'en': ['Islam']},
  'type': {'ar': ['دين'], 'en': ['Religion']},
  'DBpedia_type':
['http://dbpedia.org/ontology/EthnicGroup']},
 'كرد':
{'DBpedia_URI': ['http://dbpedia.org/resource/Kurds'],
  'label': {'ar': ['كرد'], 'en': ['Kurds']},
  'type': {'ar': ['مجموعة عرقية'], 'en': ['Ethinc group']},
  'DBpedia_type': ['http://dbpedia.org/ontology/EthnicGroup',
   'http://dbpedia.org/ontology/Band']}
}
```

The Arabic language suffers from the scarcity of databases that can be used to evaluate the performance of such tools. In order to evaluate the performance of this tool, we will use the ANERcorp dataset [37, 38] that includes five entities (Person, Location, Organization, Miscellaneous, and others). It is necessary to point out some of the difficulties facing the evaluation process, including:

The tool provided by this work is able to distinguish more than 100 entities; therefore, we must perform a manual comparison process between the type predicted by the tool and the real type.

The database contains single words. In some cases, this single word is a part of a name of an organization or a location; thus, the assigned entity will be the same entity of the full name even if the single name refers to another entity. According to the way the proposed tool works “as shown in the previous example", the entity that the tool will predict will correspond to the nature of the single word and not the full name.

Since we will do the evaluation process manually, we have selected a sample of 2000 words from the ANERcorp dataset to complete the evaluation process. Consequently, the proposed tool could annotate words belonging to more than fifty entities, but only fifty entities contain at least three words.

We measured the annotation accuracy for each entity separately; therefore, we can say that the annotation accuracy was between 75% and 100%. It should be noted here that the entities with 100% accuracy are entities that contain less than six words.

In conclusion, we can say that the tool presented by this research is comparable in its accuracy to other tools that do the same work, but it surpasses them in its ability to annotate more entities. Thus, giving a more accurate description.

VIII. Conclusion

This research aimed to reach an automatic entity annotation tools for Arabic texts. This tool uses the DBpedia database as an information source; afterwards, work was done to improve its quality in terms of the number of types that a single resource could belong to. In addition, the research focused on verifying the validity of the attribution that was made for each resource; in other words, whether the type to which the resource belongs matches the concept that this resource represents. This is with the aim of obtaining structured data from unstructured data; thus, improving the quality of the output that can be obtained through applying NLP models. The contributions made by this research were:

First, devise a new ontology from the DBpedia ontology that represents the resources with Arabic label; besides, the new ontology aims to improve the quality of automatic entity annotation by unifying the resources that belong to similar concepts within the same type;

Second, build an algorithm to verify the validity of the types that were given to the resources;

Third, build an ATEA that provides the user with the final results in a format that enables him to use them in a different field.

This tool can be developed by increasing the size of its database in two ways, the first is to use the existing resources in DBpedia but have not an Arabic label after translating their English label to the Arabic language. The second is to use a new database and expanding the classes included in the DBpedia Arabic resources ontology. Thus, it will increase the efficiency of the output of this tool; then, the possibility of using their output as inputs for classification models and cluster models.

In future works, we will develop the tool to build semantic relationships to convert text into a RDF triple store. We will use different datasets after performing all data quality checks to achieve this goal. In addition, we aim to expand languages that can be automatically annotated; consequently, this will allow the use of different texts written in different languages as an input of different NLP models like similarity model without the need for translation by the use of the information of the words annotated.

REFERENCES

[1] Völkel M, Krötzsch M, Vrandecic D, et al. Semantic wikipedia. In: Proceedings of the 15th international conference on World Wide Web. 2006, pp. 585–594.

[2] Milne D, Witten IH. Learning to link with wikipedia. In: Proceedings of the 17th ACM conference on Information and knowledge management. 2008, pp. 509–518.

[3] Auer S, Bizer C, Kobilarov G, et al. Dbpedia: A nucleus for a web of open data. In: The semantic web. Springer, 2007, pp. 722–735.

[4] Hossain BA, Salam A, Schwitter R. A survey on automatically constructed universal knowledge bases. J Inf Sci 2021; 47: 551–574.

[5] Cucerzan S. Large-scale named entity disambiguation based on Wikipedia data. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), pp. 708–716.

[6] Mihalcea R, Csomai A. Wikify! Linking documents to encyclopedic knowledge. In: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. 2007, pp. 233–242.

[7] Kulkarni S, Singh A, Ramakrishnan G, et al. Collective annotation of Wikipedia entities in web text. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. 2009, pp. 457–466.

[8] Ferragina P, Scaiella U. Fast and accurate annotation of short texts with wikipedia pages. IEEE Softw 2011; 29: 70–75.

[9] Makris C, Plegas Y, Theodoridis E. Improved text annotation with Wikipedia entities. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing. 2013, pp. 288–295.

[10] Fellbaum C, Miller G. Wordnet: An electronic lexical database (language, speech, and communication). Epub ahead of print 2000. DOI: 10.2307/417141.

[11] Brank J, Leban G, Grobelnik M. Semantic annotation of documents based on wikipedia concepts. Informatica; 42, http://www.informatica.si/index.php/informatica/article/view/2228 (2018).

[12] Strobl M, Trabelsi A, Zaiane OR. WEXEA: Wikipedia EXhaustive Entity Annotation. In: Proceedings of The 12th Language Resources and Evaluation Conference, pp. 1951–1958.

[13] Benajiba Y, Diab M, Rosso P, et al. Arabic named entity recognition: An svm-based approach. In: Proceedings of 2008 Arab international conference on information technology (ACIT). 2008, pp. 16–18.

[14] Al-Jumaily H, Martínez P, Martínez-Fernández JL, et al. A real time Named Entity Recognition system for Arabic text mining. Lang Resour Eval 2012; 46: 543–563.

[15] Yosef MA, Spaniol M, Weikum G. {AIDA}rabic A Named-Entity Disambiguation Framework for {A}rabic Text. In: Proceedings of the {EMNLP} 2014 Workshop on {A}rabic Natural Language Processing ({ANLP}). Doha, Qatar: Association for Computational Linguistics, pp. 187–195.

[16] Al-Yahya M, Al-Shaman M, Al-Otaiby N, et al. Ontology-Based Semantic Annotation of Arabic Language Text. Int J Mod Educ Comput Sci 2015; 7: 53–59.

[17] Al-Qawasmeh O, Al-Smadi M, Fraihat N. Arabic named entity disambiguation using linked open data. In: 2016 7th International Conference on Information and Communication Systems (ICICS). 2016, pp. 333–338.

[18] Albukhitan S, Alnazer A, Helmy T. Semantic Annotation of Arabic Web Documents using Deep Learning. Procedia Comput Sci 2018; 130: 589–596.

[19] Mendes PN, Jakob M, Bizer C. DBpedia: A Multilingual Cross-domain Knowledge Base. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12). European Language Resources Association, pp. 1813–1817.

[20] Bizer C, Lehmann J, Kobilarov G, et al. DBpedia-A crystallization point for the Web of Data. J web Semant 2009; 7: 154–165.

[21] Ismayilov A, Kontokostas D, Auer S, et al. Wikidata through the Eyes of DBpedia. Semant Web 2018; 9: 493–503.

[22] Lehmann J, Isele R, Jakob M, et al. DBpedia--a large-scale, multilingual knowledge base extracted from Wikipedia. Semant Web 2015; 6: 167–195.

[23] Ismail AS, Al-Feel H, Mokhtar HMO. Introducing a new arabic endpoint for DBpedia internationalization project. In: Proceedings of the 20th International Database Engineering & Applications Symposium. 2016, pp. 284–289.

[24] Zaveri A, Kontokostas D, Sherif MA, et al. User-driven quality evaluation of dbpedia. In: Proceedings of the 9th International Conference on Semantic Systems. 2013, pp. 97–104.

[25] Kontokostas D, Westphal P, Auer S, et al. Test-driven evaluation of linked data quality. In: Proceedings of the 23rd international conference on World Wide Web. 2014, pp. 747–758.

[26] Radulovic F, Mihindukulasooriya N, Garc\'\ia-Castro R, et al. A comprehensive quality model for Linked Data. Semant Web 2018; 9: 3–24.

[27] Rico M, Mihindukulasooriya N, Kontokostas D, et al. Predicting incorrect mappings: a data-driven approach applied to DBpedia. In: Proceedings of the 33rd annual ACM symposium on applied computing. 2018, pp. 323–330.

[28] Lakshen GA, Janev V, Vraneš S. Challenges in quality assessment of Arabic DBpedia. In: Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics. 2018, pp. 1–4.

[29] Jarrar M, Khalilia M, Ghanem S. Wojood: Nested {A}rabic Named Entity Corpus and Recognition using {BERT}. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference. Marseille, France: European Language Resources Association, pp. 3626–3636.

[30] Al-Thubaity A, Alkhereyf S, Alzahrani W, et al. {CA}ra{NER}: The {COVID}-19 {A}rabic Named Entity Corpus. In: Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP). Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, pp. 1–10.

[31] Public SPARQL Endpoint. DBpedia, https://wiki.dbpedia.org/public-sparql-endpoint (2020, accessed 22 July 2020).

[32] Zhang Z, Krawczyk B, Garcìa S, et al. Empowering one-vs-one decomposition with ensemble learning for multi-class imbalanced data. Knowledge-Based Syst 2016; 106: 251–263.

[33] Kahlawi A. An Ontology Driven ESCO LOD Quality Enhancement. Int J Adv Comput Sci Appl; 11. Epub ahead of print 2020. DOI: 10.14569/IJACSA.2020.0110308.

[34] Zhu L, Ghasemi-Gol M, Szekely P, et al. Unsupervised Entity Resolution on Multi-type Graphs. In: Groth P, Simperl E, Gray A, et al. (eds) The Semantic Web -- ISWC 2016. Cham: Springer International Publishing, 2016, pp. 649–667.

[35] Rahman NA, Mabni Z, Omar N, et al. A Parallel Latent Semantic Indexing (LSI) Algorithm for Malay Hadith Translated Document Retrieval. In: Berry MW, Mohamed A, Yap BW (eds) Soft Computing in Data Science. Singapore: Springer Singapore, 2015, pp. 154–163.

[36] Kahlawi A, Martelli C, Buzzigoli L, et al. A similarity matrix approach to empower ESCO interfaces for testing , debugging and in support of users ' experience. In: Pollice A, Salvati N, Spagnolo FS (eds) Riunione Scientifica della Società Italiana di Statistica -SIS. Pisa: Pearson, pp. 904–909.

[37] Benajiba Y, Rosso P, BenedíRuiz JM. ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy. In: Gelbukh A (ed) Computational Linguistics and Intelligent Text Processing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 143–153.

[38] Obeid O, Zalmout N, Khalifa S, et al. {CAM}e{L} Tools: An Open Source Python Toolkit for {A}rabic Natural Language Processing. In: Proceedings of the Twelfth Language Resources and Evaluation Conference. Marseille, France: European Language Resources Association, pp. 7022–7032.