IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 26, No. 2, April, 2021, pp. 143-149
Journal homepage: http://iieta.org/journals/isi

ARALD: Arabic Annotation Using Linked Data

Abdelghani Bouziane[1], Djelloul Bouchiha[1], Redha Rebhi[2,3], Giulio Lorenzini[4*], Noureddine Doumi[5], Younes Menni[6], Hijaz Ahmad[7]

[1] Dept. Mathematics and Computer Science, EEDIS Lab., Inst. Sciences and Technologies, Ctr Univ Naama, UDL-SBA, Naama 45000, Algeria
[2] Department of Mechanical Engineering, Faculty of Technology, University of Medea, Medea 26000, Algeria
[3] LERM - Renewable Energy and Materials Laboratory, University of Medea, Medea 26000, Algeria
[4] Department of Engineering and Architecture, University of Parma, Parco Area delle Scienze, 181/A, Parma 43124, Italy
[5] Department of Computer Science, Faculty of Technologies, University of Saida, Saida 20000, Algeria
[6] Unit of Research on Materials and Renewable Energies, Department of Physics, Faculty of Sciences, Abou Bekr Belkaid University, P.O. Box 119-13000-Tlemcen, Algeria
[7] Department of Basic Science, University of Engineering and Technology, Peshawar 25000, Pakistan

Corresponding Author Email: Giulio.lorenzini@unipr.it

https://doi.org/10.18280/isi.260201

Received: 23 Janaury 2021
Accepted: 6 April 2021

Keywords:
semantic web, linked data, linked open data, Arabic language, NLP techniques, machine learning, SPARQL, RDF, text annotation

ABSTRACT

The evolution of the traditional Web into the semantic Web makes the machine a first-class citizen on the Web and increases the discovery and accessibility of unstructured Web-based data. This development makes it possible to use Linked Data technology as the background knowledge base for unstructured data, especially texts, now available in massive quantities on the Web. Given any text, the main challenge is determining DBpedia's most relevant information with minimal effort and time. Although, DBpedia annotation tools, such as DBpedia spotlight, mainly targeted English and Latin DBpedia versions. The current situation of the Arabic language is less bright; the Web content of the Arabic language does not reflect the importance of this language. Thus, we have developed an approach to annotate Arabic texts with Linked Open Data, particularly DBpedia. This approach uses natural language processing and machine learning techniques for interlinking Arabic text with Linked Open Data. Despite the high complexity of the independent domain knowledge base and the reduced resources in Arabic natural language processing, the evaluation results of our approach were encouraging.

1. INTRODUCTION

Arabic is one of the fourth most spoken languages in the world. It covers 22 countries. So it is the official language of 422 million persons. Arabic is a very flexional and derivational language with the richest morphology [1]. Fewer efforts are made for Natural Language Processing (NLP) for the Arabic language, especially in the Linked Data context, compared to English and Latin. Work on annotating Arabic languages is a time-consuming and challenging task with limited resources, but it is beginning to emerge. The main objective of the semantic Web technology is to extend the actual Web to the semantic level and understand the textual Web by machines. Thus, the annotation of the textual data by Linked Data is crucial for developing the Web technology, especially for Arabic.

There are several problems with the natural Arabic language that could affect creating such NLP techniques and tools, especially on the Web. Morphological, grammatical and semanticized complexities, very influential or derivative language, lack of capital letters and strong ambiguity characterize the Arabic language [2]. Furthermore, English NLP tools do not meet Arabic language needs.

The semantic Web is a Web of Data, the kind of data found in databases [3]. The collection of interrelated Web datasets can also be called Linked Data, enabled by OWL, RDF, RDFS and SPARQL technologies. RDF is the base for publishing and linking data. SPARQL is the Semantic Web query language. A real-world example of Linked Open Data is the DBpedia project [4], one of the main sources of structured data on the Web. The DBpedia project builds a large-scale, multilingual knowledge base by mapping structured data from Wikipedia to the DBpedia ontology. Unfortunately, the DBpedia Arabic chapter [5] is not available online since 2017. For exploring this main ontology in the Arabic language, we use the rdfs: label property from RDF Schema Recommendation [3] that provides labels for resources in a different language, including the Arabic language.

This paper describes how our proposed system identifies entities, finds them in DBpedia and disambiguates the URIs to annotate the input text. This approach uses NLP techniques, such as tokenization, normalization, speech, tagging (pos tag) and parsing. This approach uses machine learning to implement the Named Entities Recognition (NER) module. The NER uses the Support Vector Machine (SVM) and word embedding to train the model.

The remainder of this paper is organized as follows: the related work is presented in Section 2. Section 3 is devoted to the description of the proposed system. Section 4 discusses the experimental results. Section 5 contains the conclusion of our work.

2. RELATED WORK

Many researchers have proposed various methods for annotating or converting textual information into Linked Data. Several existing approaches for entity annotation focused on annotating named entities [6, 7], which are entities of septic types (person, organization and location), or entities in a specific domain, like law, health, nutrition, food, etc. [8-10]. This section presents some works for annotating text; we focus on the Linked Data interlinking and the automatic annotation process.

Table 1. Tools for annotating Arabic text

System	Input language	Architecture	Performance
DBpedia Spotlight [11, 12]	16 languages: English, French, Deutsh, Spanish, etc.	• Spotting • Candidate Selection • Disambiguation • Filtering	**English** Accuracy: 0.851 MRR: 0.797 **French** Accuracy: 0.789 MRR: 0.677
Ref. [14]	Arabic Language	• Preprocessing • Semantic annotation • Annotation	Not available
AraTation [13]	Arabic Language	• Preprocessing • Words extraction • Semantic annotation	Precision:0.76 Recall: 0.82
AMASAT [15]	Arabic Language	• Analyzer • Matcher • Annotator	Precision: 0.86 Recall: 0.72
ARALD (our system)	Arabic Language	• Prepossessing • Named Entities Recognition • Rules-based extraction, • Semantic module, • Disambiguation	Precision: 0.91 Recall: 0.78

The most popular tool for annotating text using DBpedia is the DBpedia Spotlight presented by Mendes et al. [11] and Daiber et al. [12]. The annotation process is done using four stages: spotting, candidate selection, disambiguation and filtering. Saleh et al. [13] describe AraTation: an Arabic semantic annotation tool, the first annotating tool for the Arabic language. This tool follows architecture in three stages: (i) preprocessing to clean and normalize the text, (ii) words extraction by using a dictionary-based IE module that contains the Arabic words (in the news domain), and (iii) semantic annotation for mapping the words from the previous stage to related ontological instances. El-ghobashy et al. [14] propose a semantic annotation tool for supporting Arabic contents. The system architecture is based on preprocessing, semantic annotation and annotation module, which is used to update the knowledge base (KB) of the annotations of the user's requests. AMASAT is a semantic annotation tool for Arabic Web document released by Al-Bukhitan et al. [15]. AMASAT processes the input text by first: the analyzer module extracts textual data from the HTML document. Second, the matcher module identifies similarities between the phrases in the text and the terms of the ontology resources. Finally, the annotator module generates the annotation in the original document as embedded annotation and standalone annotation document.

Table 1 summarizes tools for annotating Arabic text with DBpedia according to their input language, the system architecture and the tool performance.

3. SYSTEM ARCHITECTURE

Our system takes an Arabic text as input and provides a labeled text with DBpedia data URIs. The architecture of the system is based on three modules: (i) candidate resources module, (ii) semantic module, and (iii) disambiguation module. First, the system receives Arabic text, split it into sentences and proceeds to the processing of each sentence using the resources extraction module to extract all candidate resources. Second, the system checks the existence of each resource in the knowledge base. Third, the system disambiguates the URIs produced in the previous step. Finally, the system produces a labeled text using Linked Data technology as can be seen in Figure 1. In the following, we describe each module and step belonging to our proposed system.

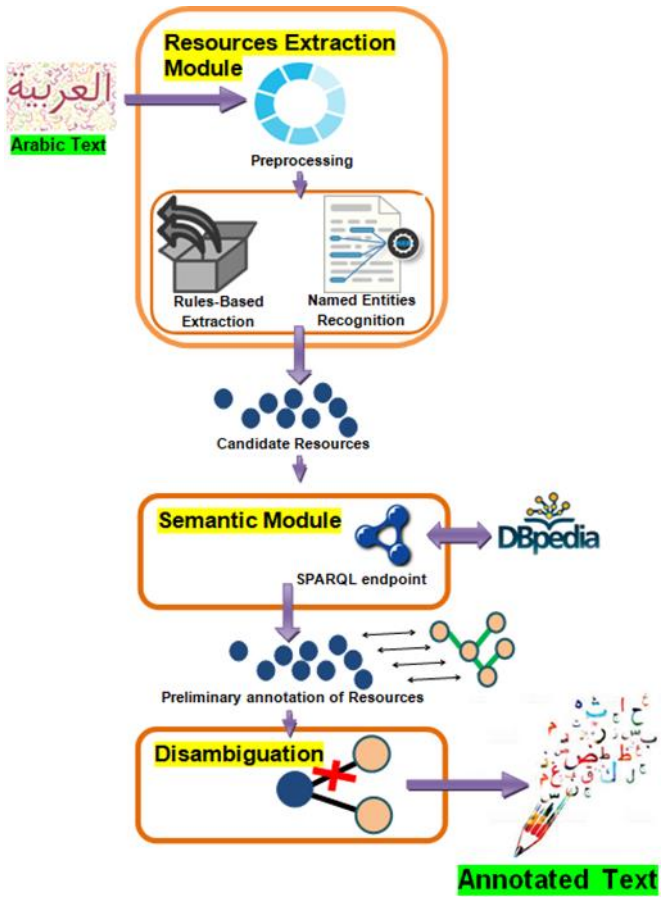

Figure 1. ARALD's architecture

3.1 Resource extraction module

This module aims to identify resources as the surface forms substrings from the input text that may be entity mentions in the DBpedia ontology. However, sentences are made up of two or more resources connected by verbs, prepositions, conjunctions, verbal phrases or nominal phrases. The set of words considered as candidate resources in the sentences are

those found in the ontology. Moreover, the existing real-world things, such as person, location and organization, are referred to as named entity mentions present in the ontology as individuals. Next, we describe the main steps for generating candidate resources.

a) Preprocessing

Tokenization and normalization are common steps in most NLP processes. Tokenization is the process of breaking down a natural language text into individual, i.e. basic sequential units. When dealing with the Arabic language, a word normalization step is required. The normalization of letters is used to correct the most common spelling errors: '|' ,'</أ/<' and 'ﺇ/|' are replaced by 'ﺍ/A', while the letter 'ﺓ/t' is replaced by 'ﻫ/h', and the letter 'ﻯ/Y' is replaced by 'ﻱ/y' (Exner and Nugues 2012). At this point, the diacritics are also removed. These errors occur when writers fail to obey Standard Arabic grammatical rules, resulting in letters written in various styles.

Note that throughout this article, every time we give an Arabic text, for readability purposes, we follow it with its Buckwalter transliteration [16] and, eventually, English translation for readability purposes.

b) Named Entities Recognition:

Developing an NE extraction application can be done using two main approaches: first, the rule-based system tries to achieve a linguistic representation by handcrafted rules by a human expert in a target domain. Second, machine learning algorithms; this approach is motivated by the evolution of the Web, which makes available a massive amount of data in different forms, structured and unstructured. This advantage increases the importance of machine learning to build predictive models effectively. A high-performance application for named entity recognition is demonstrated using machine learning and deep-learning approaches.

The Linguistic Data Consortium (LDC) releases multilingual named entity corpora, including the Arabic language. These datasets are not available for the researcher for free, but only by a costly annual license. However, the ANERcorp [17] is one of the earliest publically available named entity corpora, and it was built concerning the CoNLL standard. The ANERcorp was constructed under a coarse-grained class. Therefore, any word on the text should be annotated as one of the following tags:

B-PERS: The Beginning of the name of a PERSon;
I-PERS: The continuation (Inside) of the name of acPERSon;
B-LOC: The Beginning of the name of a LOCation;
I-LOC: The Inside of the name of a LOCation;
B-ORG: The Beginning of the name of an ORGanization;
I-ORG: The Inside of the name of an ORGanization;
B-MISC: The Beginning of the name of an entity that does not belong to any of the previous classes (MISCellaneous);
I-MISC: The Inside of the name of an entity that does not belong to any previous classes; and
O: The word is not a named entity (Other).

Therefore, a significant contribution is made by ref. [7], which release WikiFANESelective and WikiFANEWhole, an automatically built corpus for named entity using Wikipedia. The same authors [7] make available a gold standard fine-grained named entity corpora for Arabic. The taxonomy used in this work is shown in Table 2.

In this step, we use the Arabic gold-standard fine-grained NE corpora, released by Alotaibi and Lee [7], called WikiFANEgold, and drawn from Wikipedia Arabic pages, to annotate the input text. This corpus constitutes 34483 sentences with a vocabulary of 114632 words and 100 tags in the tow-level taxonomy, eight coarse-grained and 50 fine-grained.

Table 2. Taxonomy used in Alotaibi and Lee [7]

Coarse-grained Classes	Fine-grained Classes
PER: Person	Politician, Athlete, Businessperson, Artist, Scientist, Police, Religious, Engineer, Group, Other.
ORG: Organization	Government, Non-Governmental, Commercial, Educational, Media, Religious, Sports, Medical-Science, Entertainment.
LOC: Location	Address, Boundary, Water-Body, Celestial, Land-Region-Natural, Region-General, Region-International.
GPE: Geo-Political	Continent, Nation, State-or-Province, County-or-District, Population-Center, GPE-Cluster, Special.
FAC: Facility	Building-Grounds, Subarea-Facility, Path, Airport, Plant.
VEH: Vehicle	Land, Air, Water, Subarea-Vehicle, Unspecified.
WEA: Weapon	Blunt, Exploding, Sharp, Chemical, Biological, Shooting, Projectile, Nuclear, Unspecified.
PRO: Product	Book, Movie, Sound, Hardware, Software, Food, Drug, Other.

We use Support Vector Machine (SVM) to train a model for Arabic Named Entity Recognition using the dataset and taxonomy proposed by Alotaibi and Lee [18] and described above. SVMs are considered by many to be the most powerful machine learning algorithm and one of the most widely used learning algorithms today [19]. SVMs achieved the highest results in text categorization and are commonly used in NLP-related problems in various languages, including Arabic, for such methods as Named Entity Recognition and sentiment analysis.

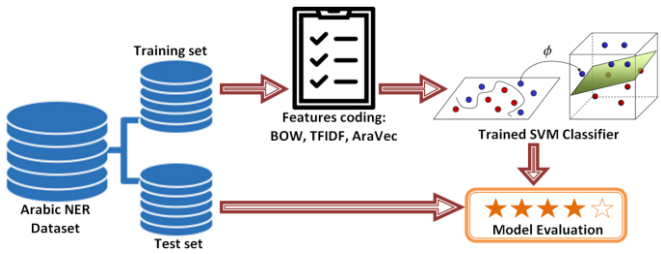

Figure 2. Named Entities Recognition based on SVM

In the overall proposed system, Figure 2 depicts our approach to recognize Arabic NE.

To address the issue of recognizing Arabic named-entities using a machine learning classifier, we need a quantitative representation of the input text. This representation is often called a vector model or feature model [20]. At this stage, we investigated three feature models:

- Bag of words: the bag-of-words (BOW) model is a representation that turns arbitrary text into fixed-length vectors by counting how many times each word appears;
- TF-IDF: TF-IDF stands for "Term Frequency - Inverse Document Frequency" model, which is a representation that transforms text into fixed-length vectors by quantifying a word in a sentence and compute a weight to each word which signifies the importance of the word in the sentences and dataset; and
- Word embedding: where words are represented in a continuous space, captures many syntactic and semantic relationships. AraVec is a pre-trained open-source word representation (word embedding) project to provide free-to-use, powerful word embedding models to the Arabic NLP research community [21]. The authors decided to collect data to build the different distributed word representation models from three completely different data sources: Twitter, World Wide Web, and Wikipedia. In our system, we use the models built from Wikipedia; 1800000 paragraphs were preprocessed to construct Wikipedia based AraVec model.

c) Rules-Based Extraction

Named entities exist mostly as instances or individuals in the input text. However, more names of concepts, classes or types must be annotated. For this need, we assume that resources in sentences can be in the form of nominal phrases or a series of different kinds of nouns, proper nouns and adjectives as shown in Table 3.

Table 3. Different grammatical forms of resources

Nominal phrase	Example
proper noun/set of proper nouns	مصطفى، محمد علي\muSTafaY, muHamadEaliy\
noun/set of nouns	قصر المرادية\qaSru AlmurAdiyap\
noun + adjective	المدينة الجديدة\Almadynapu Aljadydap\
set of nominal phrases	المدينة الجديدة سيدي عبد الله

In Linked Data, the RDF triple <subject, predicate, object> represents truth. The RDF triple's subject and object are typically called nominal phrases, and possibly classes, cases or literal values [22].

3.2 Semantic module

After extracting the candidate resources list as valuable pieces of information from Arabic sentences, we must find the reference of each resource that may correspond to a DBpedia resource. DBpedia ontology is in English, and the DBpedia Arabic chapter [5] is no longer available. For these reasons, we use the rdfs: label, which is an instance of RDF: Property. rdfs: Label provides a human-readable version of a resource's name. The Data Web has many URIs in the resource. For labeling and extending resources in the sentences, we use a SPARQL query to check the candidate resources' existence in DBpedia and interlink each one with this knowledge base. We use the following SPARQL query Listing 1 via the online service of DBpedia SPARQL endpoint:

The SPARQL endpoint response is expressed in the JSON file. JSON [23] (JavaScript Object Notation) is an open standard file format that allows data interchange using human-readable text. After parsing this JSON file, we extract text resources and URIs of the similar knowledge base.

Listing 1. SPARQL query returning the URI of an Arabic resource by using rdfs: label

```
PREFIX owl: http://www.w3.org/2002/07/owl#
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX :<http://dbpedia.org/resource/>
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX dbpedia: <http://dbpedia.org/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT ?s ?o
   WHERE { ?srdfs:label “resource”@ar.}
```

3.3 Disambiguation module

One of the most critical problems to resolve when annotating text with Linked Data is disambiguation. When matching phrases to DBpedia resources, it is likely that a single candidate resource matches multiple DBpedia entities. Moreover, the wrong candidate resource must be detected and eliminated. In fact, the disambiguation in our system is performed using Sematch [24]. Sematch is an integrated knowledge graphics framework for developing, assessing and applying semantic identity (KGs). Sematch provides several tools and datasets for similarity and enables users to calculate semantic similarities between concepts, words and entities. In this way, the conceptual similarity of KG relies on similar semantical information (i.e., path-length, depth, less common subsume and information content) and semantical similarity metrics. Concepts and entities are usually built into hierarchical taxonomies as DBpedia's ontology.

First, the system computes the semantic similarity between each concept and entity as DBpedia URI resulting from the SPARQL query execution (Eq. (1)). Second, the sum of the semantic similarity of each URI (Eq. (2)) is calculated. Third, the system computes the mean of the semantic similarity of all URIs (Eq. (3)). Fourth, we assume that the relevant URI must have the sum of the semantic similarity greater than the threshold which is the mean value computed in step three; so, we eliminate the URI which doesn't satisfy the threshold condition. Finally, we eliminate the URI, which is a part of another URI, and have the sum of the semantic similarity (sumi) less than the longer URI.

$$S_{i,j} = \text{semantic\_semelarity}(URI_i, URI_j) \quad (1)$$

$$som_i = \sum_{\substack{i=0 \\ j=i+1}}^{n} S_{i,j} \quad (2)$$

$$mean = \sum_{i=0}^{n} som_i \quad (3)$$

if $URI_i \subset URI_j$:
if $som_i < somj$:
Delete (URI_i)

3.4 Running example

As input, we give to our system the Arabic Text:

ذكرت شبكة تلفزيون سكاي نيوز أن رئيس الوزراء البريطاني بوريس "جونسون عاد إلى 10 داوننغ ستريت"

Inside the resources Extraction Module, the results are as follows:

The preprocessing output:

ذكرت شبكة تلفزيون سكاي نيوز ان رئيس الوزراء البريطاني بوريس "جونسون عاد ال 10 داوننغ ستريت"

Named Entities Recognition output:

بوريس B-PER_Politician,جونسونI-PER_Politician

سكاي B-ORG_Media, نيوزI-ORG_Media

Rules-based resources extraction output:

['تلفزيون سكاي نيوز', 'سكاي نيوز', 'رئيس الوزراء البريطاني', 'رئيس ' 'الوزراء','بوريس جونسون', '10 داوننغ ستريت','داوننغ ستريت']

Let's move on to the Semantic Module. The SPARQL query gives the following result:

['http://dbpedia.org/resource/Sky_News', '"سكاي نيوز"@ar'],
['http://dbpedia.org/resource/Prime_minister', '" رئيس الوزراء"@ar'],
['http://dbpedia.org/resource/Boris_Johnson', '" بوريس جونسون"@ar'],
['http://dbpedia.org/resource/10_Downing_Street', '"10 داوننغ ستريت"@ar'],
['http://dbpedia.org/resource/Downing_Street', '" داوننغ ستريت"@ar'],
['http://dbpedia.org/resource/Sky_(video_game_player)', '"سكاي"@ar'],
['http://dbpedia.org/resource/NEWS_(band)', '"نيوز"@ar'],
['http://dbpedia.org/resource/Johnson,_Arkansas', '"جونسون"@ar'],
['http://dbpedia.org/resource/Downing,_Missouri', '"داوننغ"@ar'],
['http://dbpedia.org/resource/Sky_News', '"سكاي نيوز"@ar'],
['http://dbpedia.org/resource/Boris_Johnson', '" بوريس جونسون"@ar']

Now, the Disambiguation module acts as follows:

Table 4 shows the similarity measurement for the resource Sky News:

Table 4. Similarity measurement for the resource Sky News

Resources	Sky_News	Sum
Prime_minister	0.153	1.51
Sky_News	1.0	2.26
Boris_Johnson	0.145	1.65
10_Downing_Street	0.291	1.06
downing_Street	0.250	1.20
Sky_(video_game_player)	0.0	0.0
NEWS_(band)	0.0	0.0
Johnson,_Arkansas	0.139	1.25
/Downing,_Missouri	0.139	0.25

The mean of the similarity measurement gives:

Mean=0.937

The resources with semantic similarity greater than the threshold (0.937) are preserved:

['http://dbpedia.org/resource/Sky_News', 2.2645417958531295],
['http://dbpedia.org/resource/Prime_minister', 1.512124381375999],
['http://dbpedia.org/resource/Boris_Johnson', 1.6587177129837716],
['http://dbpedia.org/resource/10_Downing_Street', 1.069463165549446],
['http://dbpedia.org/resource/Downing_Street', 1.2029936009961224],
['http://dbpedia.org/resource/Johnson,_Arkansas', 1.2593881109964746]]

Resources, which are part of other resources, are eliminated. Thus, we eliminate:

['http://dbpedia.org/resource/Downing_Street', 1.2029936009961224],
['http://dbpedia.org/resource/Johnson,_Arkansas', 1.2593881109964746]]

The final URIs used to annotate the Arabic Text are:

['http://dbpedia.org/resource/Sky_News', 2.2645417958531295],
['http://dbpedia.org/resource/Prime_minister', 1.512124381375999],
['http://dbpedia.org/resource/Boris_Johnson', 1.6587177129837716],
['http://dbpedia.org/resource/10_Downing_Street', 1.069463165549446],

4. EVALUATION

Supporting our approach, we implemented a tool, called ARALD, providing annotation for the Arabic language over Linked Data. Experiments have been carried out to show the efficiency of our system. Next, we give evaluation results of the Named Entities Recognition (NER) step and the overall system performance.

4.1 NER evaluation

We started with 70% of the dataset [7], described in Section 3.1.b, used as a training set and coded for the trained predictive support vector machine (SVM) model. Then, 30% of the dataset was used as a test set to see which features coding technique gives the best classification result. The metric used was the accuracy classification score that returns the correctly identified named entities' percentage. The obtained results are illustrated in the graph of Figure 3.

The results above are promising and show the power of machine learning techniques when dealing with the NER issue. This comparative study shows that, in our system the embedding AraVec model to codify words, is the most efficient choice to classify named entities compared to TFIDF and BOW techniques using the SVM learning technique.

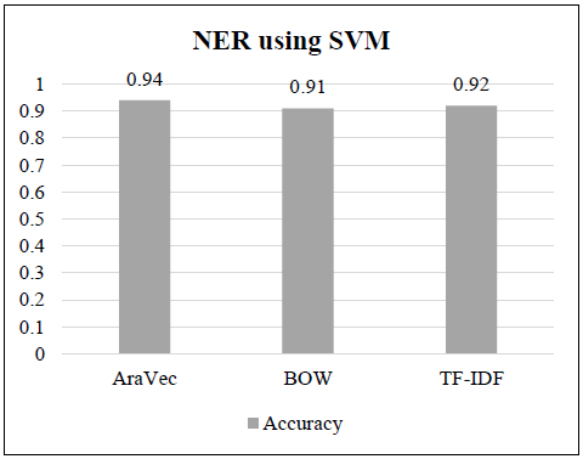

Figure 3. The success rate of the classification process using different features coding techniques

4.2 Overall system evaluation

To evaluate our system, we have done experiments using a text corpus of 100 sentences extracted from an online newspaper covering different domains; this corpus was manually annotated using DBpedia URI. Our choice is argued by the fact that there are no standard corpora for annotating Arabic texts.

Our system was evaluated using Precision, Recall (Baeza-Yates and Ribeiro-Neto 1999) and F-measure (Larsen and Aone 1999) metrics defined as follows in Table 5:

Table 5. Metrics for evaluating resources extraction and annotation processes

Metric	Definition
Precision	$\frac{\text{Correctly annotated information}}{\text{Annotated information (by our system)}}$
Recall	$\frac{\text{Correctly annotated information}}{\text{Annotated information (by a human)}}$
$F-\text{measure}$	$2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$

The results presented in Figure 4 were very encouraging since the annotation process reaches a Precision of 0.91, a Recall of 0.78, and an F-measure of 0.84.

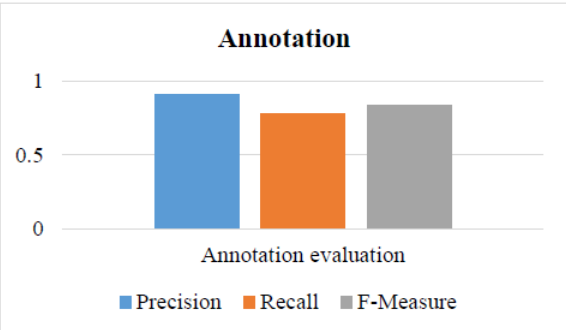

Figure 4. ARALD's evaluation results

5. CONCLUSION

This paper presents an approach for annotating Arabic texts with Linked Data, notably the DBpedia knowledge base. This constitutes the first step to enable users to link text and Web documents to the DBpedia Linked Open Data cloud through the semantic Web environment.

As an extension of the traditional Web, the semantic Web is very poor in Arabic resources. So, more effort must be made in this field to allow the Arabic language to satisfy the Arabic users' needs over the Web.

The evaluation results show the proposed system's efficacy, even in an Open Linked Data environment with no contextual domain of discourse and heterogonous vocabulary. Even if the approach achieves good results, various points must be performed such as the resource extraction, the disambiguation and the time of response.

REFERENCES

[1] Habash, N.Y. (2010). Introduction to Arabic natural language processing. Synthesis Lectures on Human Language Technologies, 3(1): 1-187. https://doi.org/10.2200/S00277ED1V01Y201008HLT010

[2] Pasha, A., Al-Badrashiny, M., Diab, M. T., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O., Roth, R. (2014). Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In LREC, 14: 1094-1101.

[3] Brickley, D., Guha, R.V., McBride, B. (2014). RDF Schema 1.1. W3C recommendation, 25: 2004-2014. http://www.w3.org/TR/2014/REC-rdf-schema-20140225/.

[4] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Sebastian, H., Mohamed, M., Patrick, V.K., Sören, A., Christian, B., Bizer, C. (2015). Dbpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. Semantic Web, 6(2): 167-195. https://doi.org/10.3233/SW-140134

[5] Al-Feel, H.A.Y.T.H.A.M. (2015). The roadmap for the Arabic chapter of DBpedia. In Mathematical and Computational Methods in Electrical Engineering, Proceedings of the 14th International Conference on Telecom. and Informatics (TELE-INFO'15), Sliema, Malta, pp. 115-125.

[6] Benajiba, Y., Rosso, P., Benedíruiz, J.M. (2007). Anersys: An Arabic named entity recognition system based on maximum entropy. In International Conference on Intelligent Text Processing and Computational Linguistics, Springer, Berlin, Heidelberg, pp. 143-153. https://doi.org/10.1007/978-3-540-70939-8_13

[7] Alotaibi, F., Lee, M. (2013). Automatically developing a fine-grained Arabic named entity corpus and gazetteer by utilizing Wikipedia. In Proceedings of the Sixth International Joint Conference on Natural Language Processing, pp. 392-400.

[8] Albukhitan, S., Helmy, T. (2013). Automatic ontology-based annotation of food, nutrition and health Arabic web content. Procedia Computer Science, 19: 461-469. https://doi.org/10.1016/j.procs.2013.06.062

[9] Stork, L., Weber, A., van den Herik, J., Plaat, A., Verbeek, F., Wolstencroft, K. (2018). From Handwritten Manuscripts to Linked Data. In International Conference on Theory and Practice of Digital Libraries, Springer, Cham, pp. 330-334. https://doi.org/10.1007/978-3-030-00066-0_34

[10] Hyvönen, E., Tamper, M., Ikkala, E., Sarsa, S., Oksanen, A., Tuominen, J., Hietanen, A. (2020). Publishing and using legislation and case law as linked open data on the Semantic Web. In European Semantic Web Conference, Springer, Cham, pp. 110-114. https://doi.org/10.1007/978-3-030-62327-2_19

[11] Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C. (2011). DBpedia spotlight: shedding light on the web of documents. In Proceedings of the 7th International Conference on Semantic Systems, pp. 1-8. Https://doi.org/10.1145/2063518.2063519

[12] Daiber, J., Jakob, M., Hokamp, C., Mendes, P.N. (2013). Improving efficiency and accuracy in multilingual entity extraction. In Proceedings of the 9th International Conference on Semantic Systems, pp. 121-124. https://doi.org/10.1145/2506182.2506198

[13] Saleh, L.M.B., Al-Khalifa, H.S. (2009). AraTation: An Arabic semantic annotation tool. In Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services, pp. 447-451. https://doi.org/10.1145/1806338.1806421

[14] El-ghobashy, A.N., Attiya, G.M., Kelash, H.M. (2014). A proposed framework for Arabic semantic annotation tool. International Journal of Computing and Digital Systems, 3(1): 45-51. http://dx.doi.org/10.12785/IJCDS/030106

[15] Al-Bukhitan, S., Helmy, T., Al-Mulhem, M. (2014). Semantic annotation tool for annotating Arabic web documents. Procedia Computer Science, 32: 429-436. https://doi.org/10.1016/j.procs.2014.05.444

[16] Buckwalter, T. (2004). Issues in Arabic orthography and morphology analysis. In Proceedings of the Workshop on Computational Approaches to Arabic Script-Based Languages, pp. 31-34.

[17] Benajiba, Y., Rosso, P. (2007). ANERsys 2.0: Conquering the NER Task for the Arabic Language by Combining the Maximum Entropy with POS-tag Information. In IICAI, pp. 1814-1823.

[18] Alotaibi, F., Lee, M. (2014). A hybrid approach to features representation for fine-grained Arabic named entity recognition. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 984-995.

[19] Ali, B.A.B., Mihi, S., El Bazi, I., Laachfoubi, N. (2020). A recent survey of Arabic named entity recognition on social media. Revue d'Intelligence Artificielle, 34(2): 125-135. https://doi.org/10.18280/ria.340202

[20] Al-Harbi, O. (2019). A comparative study of feature selection methods for dialectal Arabic sentiment classification using support vector machine. arXiv preprint arXiv:1902.06242.

[21] Soliman, A.B., Eissa, K., El-Beltagy, S.R. (2017). Aravec: A set of Arabic word embedding models for use in Arabic NLP. Procedia Computer Science, 117: 256-265. https://doi.org/10.1016/j.procs.2017.10.117

[22] AlAgha, I., Abu-Taha, A. (2015). AR2SPARQL: An Arabic natural language interface for the semantic web. International Journal of Computer Applications, 125(6): 19-27.

[23] Nurseitov, N., Paulson, M., Reynolds, R., Izurieta, C. (2009). Comparison of JSON and XML data interchange formats: a case study. Caine, 9: 157-162.

[24] Zhu, G., Iglesias, C.A. (2017). Sematch: Semantic similarity framework for knowledge graphs. Knowledge-Based Systems, 130: 30-32. https://doi.org/10.1016/j.knosys.2017.05.021