Different valuable tools for Arabic sentiment analysis: a comparative evaluation

Youssra Zahidi[1], Yacine El Younoussi[2], Yassine Al-Amrani[3]
[1,2]Information System and Software Engineering Laboratory, Abdelmalek Essaadi University, Morocco
[3]Technologies de l'Information et Modélisation des Systèmes, Abdelmalek Essaadi University, Morocco

Article Info

Article history:

Received Oct 9, 2019
Revised Aug 11, 2020
Accepted Aug 26, 2020

Keywords:

ANLP
ASA
ASA programming languages
Java libraries
Python libraries

ABSTRACT

Arabic Natural language processing (ANLP) is a subfield of artificial intelligence (AI) that tries to build various applications in the Arabic language like Arabic sentiment analysis (ASA) that is the operation of classifying the feelings and emotions expressed for defining the attitude of the writer (neutral, negative or positive). In order to work on ASA, researchers can use various tools in their research projects without explaining the cause behind this use, or they choose a set of libraries according to their knowledge about a specific programming language. Because of their libraries' abundance in the ANLP field, especially in ASA, we are relying on JAVA and Python programming languages in our research work. This paper relies on making an in-depth comparative evaluation of different valuable Python and Java libraries to deduce the most useful ones in Arabic sentiment analysis (ASA). According to a large variety of great and influential works in the domain of ASA, we deduce that the NLTK, Gensim and TextBlob libraries are the most useful for Python ASA task. In connection with Java ASA libraries, we conclude that Weka and CoreNLP tools are the most used, and they have great results in this research domain.

Corresponding Author:

Youssra Zahidi,
Information System and Software Engineering Laboratory,
Abdelmalek Essaadi University, Tetuan, Morocco.
Email: youssra1994zahidi@gmail.com

1. INTRODUCTION

Natural language processing (NLP) is a subfield of computer science, linguistics, artificial intelligence, and information engineering interested by the interactions between human (natural) languages and computers, in particular how to program computers to treat and process a massive quantity of natural language data. Arabic natural language processing ANLP tries to build software eligible to treat Arabic linguistic data automatically for a specific application. The Arabic language is recognized as the 4th most used language of the Internet. It is the formal language of twenty-two countries, spoken by more than four hundred million speakers. It is a Semitic language that is characterized by its literary abundance. Arabic morphology is rich, complex, and highly ambiguous. For this reason, it poses a variety of problems in the field of NLP. Nowadays, ANLP has obtained significant value. A large variety of applications have been built like: sentiment analysis [1, 2], machine translation, question answering, named entity recognition, etc. These applications must adapt to the complicated structure of Arabic [3]. This Semitic language has its own special features; for example, it has no capitalization; the Arabic alphabet contains 29 consonants and 11 vowels. Moreover, the Arabic language is written from right to left, and its letters change format depending on their place in the word.

ANLP-related problems [4, 5] can be summarized as follows: To begin with, the problem of multiple vowellation and the complexity of the Arabic graphic word structure (an Arabic descriptive word can correspond to a whole sentence in French). Besides, the word order is relatively free in an Arabic sentence (verb + object + subject (VOS); verb + subject + object (VSO); object + verb + subject (OVS)). Arabic does not contain capital letters unlike most Latin languages. This makes ANLP, such as recognition of entity names, very difficult. Another distinctive feature of Arabic is diacritical marks (short vowels): the same word with different diacritics can express different meanings. Diacritics are usually omitted, causing ambiguity. Also, this Semitic language is very inflectional and derivative, which makes its morphological analysis a complex task. It is derivative in that all the Arabic words have a three- or four-character root verb, and it is inflectional because each word consists of zero or more affixes (prefix, infix and suffix) and a root. As a result, the lack of Arabic resources, such as libraries, that support the Arabic language corpora makes ANLP research more challenging. For this reason, Arabic is more complex and difficult to process in NLP field compared to the other famous languages,

Applications of ANLP are widely spread because people communicate almost everything in language [6]. Among these applications, we focus on Arabic sentiment analysis in our research study. ASA or opinion mining aims at defining the attitude, the sentiment polarity (positivity, neutrality, negativity) of a writer or another subject concerning a particular event [7, 8]. A large number of sentiments are borne in posts on many social media platforms like (Twitter, Facebook, YouTube, Instagram). Sentimental Analysis is performed using various machine learning techniques [9-11, statistical models, and NLP for feature extraction from extensive data.

Sentiment analysis has various trending applications in many fields. In politics, it can aid in inferring the free orientation and reaction towards political events, which helps in decision making. In business, it permits companies to automatically collect their customers' opinions on their services [12]. Sentiment analysis can be done at several levels, document level [13], sentence-level, and subject-level.

In this research project, which relies on the domain of ASA, we try to do a comparative evaluation to conclude the most valuable programming languages, which are abundant at the level of ASA libraries. We compare these libraries to deduce the most powerful ones. When we talk about the English language, for example, there are various NLP tools advantageous for various NLP tasks, especially in sentiment analysis SA. Nevertheless, this is not the same situation for the Arabic language. Due to its ambiguity, syntactic and morphological abundance and richness, the Arabic language is deemed as the most difficult language. There are a limited number of libraries that support it. This complex nature, with the lack of its resources and the diversity of dialects, imposes difficulties on the development in the field of ASA research. Choosing the most appropriate group of libraries that meets our specific needs is very difficult and imposes an in-depth evaluation. To solve this major problem, we rely on a variety of valuable aspects in this comparative evaluation. This comparative evaluation is critical, in that it would enable various researchers who are interested in using ASA in their projects to build appropriate decisions about available libraries that meet their requirements and needs accurately.

The rest of the paper is described like this: the second section emphasizes ASA programming languages and their famous Libraries. Section 3 offers our in-depth comparison between the most useful Java and Python libraries for ASA. The results are debated in detail in section 4, and this work is finished with final thoughts in section 5.

2. APPLIED PROGRAMMING LANGUAGES AND LIBRARIES

Various programming languages are applied in Deep Learning and ANLP (like C++, R, Perl, Prolog, Lisp ...). These, however, are known by their scarcity of appropriate groups of libraries used in modern applications in these domains. Nowadays, various Python and Java libraries have been built to cater to the requirements and needs in current Deep Learning and ANLP modern tasks. In view of these tools' abundance, reputation, and high performance, Java and Python can be considered as the most widely used programming languages in these domains. That is why we are basing on Java and Python programming languages in this in-depth evaluation, as they are the most commonly used. We will be beginning by choosing appropriate programming languages to conclude what are identify their most potent libraries in the ASA domain.

2.1. Arabic sentiment analysis using Python

Python is a powerful programming language with excellent functionality and feature for processing natural language, its semantics and syntax are transparent, and it has excellent string-handling functionality. As an object-oriented language, Python permits methods and data to be encapsulated and reused easily. As an interpreted language, Python facilitates interactive exploration. As a dynamic language, Python allows variables to be typed dynamically and add attributes to objects on the fly, facilitating rapid development [14]. Many researchers recommend this powerful programming language. For instance: Steven Bird and Edward Loper in [14] strongly recommend the use of Python in NLP projects, and through their work [15], they deduce that this programming language is the best, providing a large variety of benefits.

In his paper [16], Nitin Madnani chose to employ Python because he confirms that this programming language has a large variety of benefits over the other programming languages, such as an easy-to-use object-oriented paradigm, high readability, strong Unicode support, easy extensibility, and a powerful standard library. It is very efficient and has been applied in complex and difficult NLP projects.

The Theano Development Team encourages the use of Python through this work [17], which they consider a flexible programming language providing a straightforward manner to react with data and allowing for fast prototyping. Moreover, paper [18] offers a critical assessment of existing Python infrastructure for NLP new tasks. Through their case study: Automatic Aspectual Classification of Verbs in an Untagged Corpus, the authors found that Python's core libraries offer perfect coverage of essential machine learning algorithms.

Python is especially more appropriate for various reasons: free and simple, object-oriented, and compatible with so many platforms, a large number of libraries for Python. Nevertheless, we also have to know the downsides of choosing it over another programming language: Speed limitations, Weak in mobile computing, and browsers.

2.1.1. Python libraries

It is fundamental first to show the most useful Python libraries that have been proven in the domain of ASA: NLTK, TextBlob, and Gensim.

a. NLTK: is a leading platform for NLP. A set of core modules (libraries and programs) offers basic data types that are utilized throughout the tool. NLTK is a perfect starting point for researchers and students in the domain of NLP because of its numerous benefits. That is why NLTK has been named "a wonderful tool for teaching and working in computational linguistics using Python" and the "mother" of all NLP libraries. The significant advantage of using NLTK is that it is entirely self-contained. Not only does it provide suitable functions that can be used as building blocks for common NLP tasks. This group of applications and libraries from the University of Pennsylvania has earned considerable traction in Python-based SA systems since its conception in 2001.

b. TextBlob: it is a python library for processing textual data; it provides a simple API to access its methods and do basic NLP tasks such as sentiment analysis, part-of-speech tagging, classification, translation... The sentiment function of TextBlob returns two properties, subjectivity, and polarity. Subjective sentences usually refer to personal opinion, judgment, or emotion, whereas objective refers to factual information. Subjectivity is also a float which lies in the range of [0,1]. Polarity is a float that lies in the range of [-1,1] where 1 means a positive statement and -1 means a negative statement.

c. Gensim: it is an open-source library for unsupervised subject modeling and NLP, using modern statistical machine learning. It is considered as a robust vector space modeling tool implemented in Python. Contrary to NLTK, Gensim is the best way to process massive datasets. Gensim library was primarily built for document similarity estimation, and this treatment is the most developed in the package. It supports three main NLP modern tasks: retrieve semantically similar documents, scalable statistical semantics, and analyze plain-text documents for semantic structure [18]. Gensim includes streamed parallelized implementations of many algorithms like fastText, word2vec, and doc2vec that are used a lot in the field of Arabic sentiment analysis. Its highly and native optimized implementation of Google's word2vec machine learning models makes it a strong contender for inclusion in a SA project, either as a core framework or as a library resource.

In Table 1, we try to highlight many advantages and disadvantages of the most used Python libraries in Arabic sentiment analysis.

2.2. Arabic sentiment analysis using Java

Because of its best features, Java is a powerful programming language for performing NLP. The Java application, like just in time, processes a large quantity of data as rapidly as possible. The multi-threading characteristic of Java is very significant for the heavily loaded application. This application is useful in NLP in that the task is divided into several threads, thus reducing the time.

NLP stored a wide variety of linguistic files. Java has an excellent ability to store data without any changing a single code. The Java database connectivity API serves as a bridge between Java application and the database. The linguistic knowledge updated without changing the single line of Java code, and it stored in the database. In [19], the authors strongly recommended the use of Java programming language. Besides, the authors of [20] found that Java is the best and the most useful programming language.

The following section presents several benefits of Java: it is simple, secure, interpreted, distributed, object-oriented, platform-independent, and multi-threaded. According to sun microsystems, Java has the following essential strengths: security, portability, ease of use, robustness, and distributed process across the Web. There is, however, the scope for Java improvement as it continues to have some disadvantages: Java can be seen as significantly slower and more memory-intensive than natively compiled languages, the single paradigm language, Look and feel. The default feel and look of GUI applications written in Java using the Swing tool are very different from native applications.

2.2.1. Java libraries

In this section, we will show the most powerful Java library for ASA: Weka, CoreNLP, and Gate.

a. The Stanford CoreNLP: it offers a set of human language technology tools. It is a Java annotation pipeline framework that provides language processing tasks and offers most of the common essential NLP steps, from tokenization through to co-reference resolution [21]. Stanford CoreNLP's purpose is to make it simple to apply a bunch of linguistic analysis tools to a text. This library is built to be highly flexible and extensible. The most supported language is the English language, but other languages, like Arabic, German, Chinese, Spanish, and French, are also available. Its features, relative ease of implementation, dedicated SA tools, and excellent community support make CoreNLP a severe contender to production, even if its Java-based architecture could entail a little extra engineering and overhead, in certain circumstances. The Stanford NLP library can be used using Python because there are several packages and interfaces for using Stanford CoreNLP in Python (independent of NLTK).

b. Weka: it is open-source software available under the GNU general public license. It is an accessible suite of machine learning software written in Java, developed at the University of Waikato, New Zealand. The Weka workbench includes a group of algorithms and visualization tools for predictive modeling and data analysis, with graphical user interfaces for easy access to this functionality. WEKA was used to perform sentiment classification to solve problems in various fields. It has been used for SA purposes by a large variety of researches and papers.

c. Gate: it is an open-source and Cross-platform Java software toolkit capable of resolving all text processing problems. It contains a nearly-new information extraction system "ANNIE," which is a group of modules containing a named entities transducer, a part-of-speech tagger, a gazetteer, a tokenizer, a co-reference tagger, and a sentence splitter. This library supports various languages: Arabic, English, French, German, Chinese, Italian, Bulgarian, Romanian, Hindi, Cebuano, Romanian, Danish, and Russian. There are some valuable Gate plugins that are very useful in Arabic sentiment analysis, such as SEAS and SAGA.

Table 1. Comparison of the most used Python libraries in ASA

Library	Advantages	Disadvantages
NLTK	- Support the most significant number of languages compared to other libraries. - The Most Well-Known and full NLP library. - Many third-party extensions. - Fast sentence tokenization. - Plenty of approaches to each NLP task	- Quite slow. - It is complicated to learn and use. - Processes strings which are not very typical for object-oriented language Python. - In sentence tokenization, NLTK only splits text by sentences, without analyzing the semantic structure.
Gensim	- Provides tf-idf vectorization, word2vec, document2vec, latent semantic analysis, latent Dirichlet allocation. - Works with large datasets and processes data streams. - Supports deep learning.	- Does not have enough tools to provide full NLP pipeline, so should be used with some other library (Spacy or NLTK) - Designed primarily for unsupervised text modeling
TextBlob	- Offers language translation and detection which is powered by Google Translate - Simple to apply and intuitive interface to NLTK library	- Slow - No integrated word vectors - No neural network models

3. COMPARATIVE STUDY OF ASA LIBRARIES

In our in-depth comparative study, we try to choose the most valuable group of libraries that meets our needs relying on a variety of valuable aspects and levels:

3.1. Comparative study of the most potent ASA libraries based on the literature

In Table 2, we try to highlight numerous characteristics of Arabic sentiment analysis libraries and famous works based on the literature. According to the literature, we concluded that NLTK, Weka, Gensim, TextBlob, and Stanford CoreNLP libraries are beneficial compared to other famous Libraries in the field of ASA and we found many Articles which adopted the use of NLTK, Weka, TextBlob and Gensim libraries in their works more than Stanford CoreNLP and Gate libraries.

Table 2. Comparison of the most potent ASA libraries

Library	License	Platform	Highlights	Works in ASA
NLTK	Apache 2.0	Cross-platform	Massive numbers of languages and tools supported; well-developed community and documentation	[22][23][24][25][26][27][28][29][30][31][32]
Gensim	LGPL	Windows, Linux, Mac OS X and should works on any other platform that supports Python 2.6+ and NumPy	Speedy, scalable, strong native capabilities; commercial spinoffs available	[33][34][35][36][37][23] [32]
TextBlob	MIT License	Cross-platform	This library stands on the giant shoulders of NLTK and pattern and plays nicely with both. therefore, making it easy for beginners by providing an intuitive interface to NLTK	[38][39][40][41][42][43] [44]
Stanford CoreNLP	GNU GPL	Cross-platform	Platform-agnostic; multi-language support; a live demo available	[25] [45] [35] [46]
Weka	GNU GPL	IA-32, x86-64; Java SE	Portability, since it is completely implemented in Java and thus works on almost any new computing platform. A large group of modeling techniques and data preprocessing. Simple of use due to its graphical user interfaces.	[47][48][49][50][51][52][53][54][55][56][57]
Gate	LGPL	Cross-platform	Gate has plugins for machine learning with Weka, MAXENT, SVM light, RASP, and fast LibSVM integration, a perception implementation for managing ontologies like WordNet, plugins for querying search engines like Yahoo or Google, and plugins for PoS tagging with Brill or TreeTagger.	[58] [59] [60]

3.2. Comparative study of open software libraries based on the community on GitHub results

GitHub provides plans for both free accounts and private repositories, which are commonly applied to host open-source projects. It is the biggest host of source code in the glob. The numbers in the GitHub site are permanently variable. That is why we will designate the visitation date of these pieces of information (10/06/2020). Table 3 shows the GitHub results. Through the results, we deduce that Gensim and NLTK are the most applied, pursued by CoreNLP, TextBlob, Weka, and lastly, GATE.

Table 3. GitHub results

Library	NLTK	Gensim	TextBlob	CoreNLP	Weka	Gate
Language	Python	Python	Python	Java	Java	Java
Stars	8,975	10,867	7,084	7,254	302	105
Forks	2,346	3,800	942	2,400	240	139
Contributors	291	336	22	100	1	44
Commits	13,888	3,928	537	16,022	9,612	3.403

3.3. Comparative study of the open software libraries based on multiple criteria

In Table 4 (see appendix), we try to show various criteria of NLP Tools like the Documentation, Characteristics, also, the supported treatments of each NLP library, i.e., NLTK, Genism, TextBlob Python Libraries and CoreNLP, Weka, GATE Java Libraries. We will make a comparison between these libraries to reach a conclusion on the most potent libraries that meets our needs very well.

4. RESULTS AND DISCUSSION

In this comparative study, we try to adopt two major matters. The first is that numerous researchers are confounded about what programming language they have to apply for various ANLP modern tasks, especially for the Arabic sentiment analysis field. The second issue is that there are a large variety of NLP libraries, which is why many researchers find it very hard to select a suitable set of libraries in their ASA research projects and which ones meet their needs best. For this reason, they use ANLP libraries for their ASA research projects, but without justifying their option. Both matters are debated in more detail below:

4.1. Selecting the suitable programming language

Among various programming languages (such as C++, R, Perl, Prolog, Lisp...), we selected Java and Python programming languages in our study. This choice is justified by their broad popularity usefulness and importance for current ANLP tasks, especially for the Arabic Sentiment Analysis domain. Besides, these two programming languages have a large variety of powerful libraries in the ASA field.

4.2. Choosing an adequate library for Arabic sentiment analysis project

Thanks to the diversity of available NLP libraries, most researchers use various libraries in their research projects without explaining the cause behind this use. We aimed to rely on our review of the literature on the most potent and useful ASA libraries, namely: NLTK, Genism, TextBlob, CoreNLP, Weka, and Gate. The choice of the library relies on the specific problem you are dealing with in Sentiment Analysis. We can use each of them in various scenarios. we tried to give you a general summarize of them, and we hope it can help you make the right option for your problem:

a. NLTK: is very useful. If you know to program in Python, then NLTK is a smart choice as it contains the functionalists of Stanford CoreNLP and Weka Tools. Other than this, you can benefit from lexical resources with ease, such as WordNet, often indispensable in the domain of ASA. Such as CoreNLP, NLTK provides various wrappers for many programming languages and comes with a variety of resources.
b. Gensim: its highly and native optimized implementation of Google's word2vec machine learning models makes it a strong candidate for inclusion in a sentiment analysis project, either as a library resource or as a core framework. Contrary to NLTK, Genism is a great option for processing massive datasets. At the same time, it does not accept a significant number of current NLP tasks such as NLTK.
c. TextBlob: it is relied on NLTK and Pattern. It has an excellent API for all the common NLP treatments. It is a more practical library focused on everyday usage. It is perfect for initial prototyping in almost every NLP project. Unfortunately, it inherits the low performance from NLTK, and therefore it is not suitable for large scale production usage. Many researchers considered TextBlob Library as one of Python's libraries to execute Sentiment Analysis.
d. StanfordCoreNLP: it is helpful if you need part of speech categories, co-reference, or named entities in text. These have been employed as potential features by the sentiment analysis research community. The Stanford CoreNLP is one of the most potent libraries among a large variety of great NLP libraries because it is easily comprehensible. Compared to other libraries, CoreNLP is easy to set up and run since users do not need to understand complex installations and procedures, and its users only require to have a little background of pieces of information about Java before they can get started.
e. Weka: it is useful if we already hold data with each data point holding a feature vector, then we can employ this tool for clustering our data. Helpful if we also hold the gold predicted outputs for our data, we can build classifiers. Simple to employ GUI accessible and highly configurable.
f. Gate: it is advantageous if we want to create a pipeline. Developers contribute language analysis modules for various languages that are available to be used plugged into your pipeline. Helpful if you have a new approach, you can write a customized module in JAVA and plug it into the pipeline, and a complete system will be obtainable.

As a conclusion of this part, each library has its advantages to ANLP Tasks, and each one was built to meet the researcher's purposes. Our inference raises two main parts :

a. The first one is to do with ANLP programming languages Python and Java, which are very popular in the ANLP domain. However, we recommend Python because it is less complicated than Java, it has powerful and valuable ANLP libraries compared to Java, and Through our comparative study, we conclude that the most valuable, robust and used ASA Libraries (NLTK, Gensim, and TextBlob) belong to Python programming language. For this reason, we will adopt Python in order to accomplish our ASA research project easily and perfectly.
b. The second point relates to ANLP libraries, which are all very useful. However, according to the literature and large variety of powerful and significant works in the domain of ASA, we conclude that the NLTK, Gensim, and TextBlob libraries are the most used for Python ASA task because they have numerous advantages compared to other ANLP libraries. As for the Java ASA tools, we find that Weka and CoreNLP tools are the most used and famous, and they have great results in this field.

5. CONCLUSION

Because of their popularity and large abundance in libraries for the ANLP domain, we selected Java and Python programming languages in our comparative study. In this work, we described a variety of ANLP tools which are considered as the most powerful and used. However, there are other tools in other programming languages that also could be very helpful and useful. Besides, we have tried to evaluate the various libraries using several aspects and multiple criteria.

It can be deduced that each programming language has its benefits and advantages, each library also has its characteristics for ANLP new tasks, and each one was built to meet the researcher's purposes. Therefore, it is tough to select the best Arabic NLP libraries because there is not only one single aspect or criterion to do this. The selection of the most suitable libraries depends on the research project and which part of the ANLP field is concerned. For this reason, we relied on our work, which deals with the domain of ASA in order to select its most potent and useful libraries with ease.

APPENDIX

Table 4. NLP Libraries' Comparison

Variable	Characteristics	Supported treatments
NLTK (Python)	- It is the "origin" of all NLP libraries. - Very good for the de-facto standard for various NLP tasks and educational purposes. - It offers an extensible, simple, uniform framework for projects, class demonstrations, and assignments. It is well documented, simple to learn, and easy to apply.	Accessing corpora, string processing, collection discovery, tokenization, stemming, POS tagging, chunking, named entities identification, semantic interpretation, classification, probability estimation, evaluation metrics, translation, dependency parsing, automatic summarization, sentiment analysis, language modeling, twitter processing, logical semantics.
Gensim (Python)	- A subject modeling toolkit implemented in Python. - It applies SciPy, optionally Cython, and NumPy for performance. - Scalability - Very efficient implementations - Converters & I/O formats - Fast: Robust - Similarity queries	- Gensim was originally designed for estimating document similarity, and this feature is the most sophisticated of the package. - Evolutive statistical semantics; - Analyze plain-text documents for semantic structure; - Recover semantically similar documents;
TextBlob (Python)	Examples of NLP TextBlob Quickstart use cases: - Sentiment Analysis - Spelling Correction - Translation and Language Detection	Tokenization, NER, POS, classification, sentiment analysis, parsing, spellcheck, language detection, and translation
CoreNLP (Java)	- An integrated NLP tool with a wide range of grammar analysis tools; - A robust annotator for arbitrary texts, widely applied in production; - A regularly updated package, with the highest quality text analytics support for several major (human) languages; - Available APIs for most significant new programming languages; - Ability to run as an easy web service.	Sentiment analysis, information extraction, named entity recognition, part-of-speech tagging, co-reference resolution system, parsing, bootstrapped pattern learning
Weka (Java)	- Portable and simple to apply. - Adapted to make new ways to machine learning designs - Latest trends in artificial intelligence - Free online courses available - Extremely resourceful books and publications available - Highly educated, skilled and committed professors	Sentiment Analysis data preprocessing, clustering, regression, classification, visualization, and feature selection.
Gate (Java)	- SEAS (Gate plugin): is a set of processing and linguistic resources, written in Java, developed to run sentiment and emotion analysis over text using the GATE platform. Because of the nature of GATE, the text format should be plain or XML. The sentiment analysis modules are executed in embedded inside SEAS. - SAGA (Sentiment and Emotion Analysis integrated into GATE) is a set of processing and linguistic resources, written in Java, developed to run sentiment and emotion analysis over text using the GATE platform. SAGA is distributed as a GATE plugin.	Sentiment analysis, information extraction, part-of-speech tagging, sentence segmentation, named entity recognition, tokenization, co-reference tagging,

REFERENCES

[1] N. S. Reddy, B. Prabadevi, and B. Deepa, "Heart rate encapsulation and response tool using sentiment analysis," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 4, pp. 2585–2592, Aug. 2019.

[2] K. V. Ghag and K. Shah, "Conceptual sentiment analysis model," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 4, pp. 2358–2366, Aug. 2018.

[3] A. Farghaly and K. Shaalan, "Arabic natural language processing: Challenges and solutions," *ACM Trans. Asian Lang. Inf. Process.*, vol. 8, no. 4, pp. 1–22, Dec. 2009.

[4] H. Abdelnasser *et al.*, "Al-Bayan: An Arabic Question Answering System for the Holy Quran," in *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pp. 57–64, 2014.

[5] W. Bakari, P. Bellot, and M. Neji, "AQA-WebCorp: Web-based Factual Questions for Arabic," *Procedia Comput. Sci.*, vol. 96, pp. 275–284, 2016.

[6] Y. Zahidi, Y. El Younoussi, and C. Azroumahli, "Comparative Study of the Most Useful Arabic-supporting Natural Language Processing and Deep Learning Libraries," in *2019 International Conference on Optimization and Applications, ICOA 2019*, 2019.

[7] H. G. Hassan, H. M. Abo Bakr, and I. E. Ziedan, "A framework for Arabic concept-level sentiment analysis using SenticNet," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 5, pp. 4015–4022, Oct. 2018.

[8] A. Alrumaih, A. Al-Sabbagh, R. Alsabah, H. Kharrufa, and J. Baldwin, "Sentiment analysis of comments in social media," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 6, pp. 5917-5922, Dec. 2020.

[9] Y. Al-Amrani, M. Lazaar, and K. E. Elkadiri, "Sentiment analysis using supervised classification algorithms," in *ACM International Conference Proceeding Series*, vol. Part F1294, 2017.

[10] Y. Al-Amrani, M. Lazaar, K. Eddine, and E. L. Kadiri, "Sentiment analysis using hybrid method of support vector machine and decision tree," *J. Theor. Appl. Inf. Technol.*, vol. 95, no. 7, pp. 1886-1895 2018.

[11] Y. Al Amrani, M. Lazaar, and K. E. El Kadirp, "Random forest and support vector machine based hybrid approach to sentiment analysis," in *Procedia Computer Science*, vol. 127, pp. 511–520, 2018.

[12] Y. Al Amrani, M. Lazaar, and K. E. El Kadiri, "A novel hybrid classification approach for sentiment analysis of text document," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 6, pp. 4554–4567, 2018.

[13] Y. Al Amrani, M. Lazaar, and K. E. El Kadiri, "Recovery of the opinions through the specificities of documents text," in *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems, WITS 2019*, 2019.

[14] K. Ewan and E. Loper, *Natural Language Processing with Python*, 1st. ed. O'Reilly Media, Inc., 2009.

[15] S. Bird and E. Loper, "NLTK: The Natural Language Toolkit," in *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, 2004, pp. 214–217.

[16] N. Madnani, "Getting started on natural language processing with Python," *Crossroads*, vol. 13, no. 4, pp. 5–5, 2007.

[17] The Theano Development Team *et al.*, "Theano: A Python framework for fast computation of mathematical expressions," *CoRR*, May 2016.

[18] A. Drozd, A. Gladkova, and S. Matsuoka, "Python, performance, and natural language processing," in *Proceedings of PyHPC 2015: 5th Workshop on Python for High-Performance and Scientific Computing - Held in conjunction with SC 2015: The International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–10, 2015.

[19] T. Read, E. Bhrcena, and P. Faber, "Java and its role in Natural Language Processing and Machine Translation," *Proc. Mach. Transl. Summit V1*, pp. 224–231, 1997.

[20] A. Pinto, H. G. Oliveira, and A. O. Alves, "Comparing the performance of different NLP toolkits in formal and social media text," in *OpenAccess Series in Informatics*, vol. 51, pp. 31–316, 2016.

[21] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. Mcclosky, "The Stanford CoreNLP Natural Language Processing Toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60, 2014.

[22] A. A. Aliane, H. Aliane, M. Ziane, and N. Bensaou, "A genetic algorithm feature selection based approach for Arabic Sentiment Classification," in *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, 2017.

[23] G. Badaro *et al.*, "EMA at SemEval-2018 Task 1: Emotion Mining for Arabic," in *Proceedings of The 12th International Workshop on Semantic Evaluation*, pp. 236–244, 2018.

[24] M. Alfonse and E.-S. M. El-Horbaty, "Opinion Mining for Arabic Dialects on Twitter," *Egypt. Comput. Sci. J.*, vol. 42, no. 4, pp. 52–61, 2018.

[25] S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 Task 4: Sentiment Analysis in Twitter," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 502–518, 2017.

[26] L. Al-Horaibi, M. Badruddin Khan, and L. Al-Horaibi Muhammad Badruddin Khan, "Sentiment Analysis of Arabic Tweets Using Semantic Resources," *Int. J. Comput. Inf. Sci.*, vol. 13, no. 1, 2017.

[27] N. El-Naggar, Y. El-Sonbaty, and M. Abou El-Nasr, "Sentiment analysis of modern standard Arabic and Egyptian dialectal Arabic tweets," in *Proceedings of Computing Conference 2017*, vol. 2018-Janua, pp. 880–887, 2018.

[28] A. Mourad and K. Darwish, "Subjectivity and Sentiment Analysis of Modern Standard Arabic and Arabic Microblogs," in *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 55–64, 2013.

[29] D. Gamal, M. Alfonse, E.-S. M. El-Horbaty, and A.-B. M. Salem, "Twitter Benchmark Dataset for Arabic Sentiment Analysis," *Mod. Educ. Comput. Sci.*, vol. 1, pp. 33–38, 2019.

[30] H. Mulki, H. Haddad, M. Gridach, and I. Babaoğlu, "Tw-StAR at SemEval-2017 Task 4: Sentiment Classification of Arabic Tweets," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 664–669, 2017.

[31] D. Suleiman and A. Awajan, "Comparative Study of Word Embeddings Models and Their Usage in Arabic Language Applications," in *ACIT 2018 - 19th International Arab Conference on Information Technology*, 2019.

[32] A. A. Altowayan and L. Tao, "Word embeddings for Arabic sentiment analysis," *Proc. - 2016 IEEE Int. Conf. Big Data, Big Data 2016*, pp. 3820–3825, 2016.

[33] A. Barhoumi, Y. Estève, C. Aloulou, and L. H. Belguith, "Document embeddings for Arabic Sentiment Analysis," in *Proceedings of the First Conference on Language Processing and Knowledge Management*, 2017, vol. 1988.

[34] A. Aziz Altowayan and A. Elnagar, "Improving Arabic sentiment analysis with sentiment-specific embeddings," in *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, vol. 2018, pp. 4314–4320, 2018.

[35] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "Improving Sentiment Analysis in Arabic Using Word Representation," in *2nd IEEE International Workshop on Arabic and Derived Script Analysis and Recognition, ASAR 2018*, pp. 13–18, 2018.

[36] S. Medhaffar, F. Bougares, Y. Estève, and L. Hadrich-Belguith, "Sentiment Analysis of Tunisian Dialects: Linguistic Ressources and Experiments," in *Proceedings of the Third Arabic Natural Language Processing Workshop*, pp. 55–61, 2017.

[37] A. El-Kilany, A. Azzam, and S. R. El-Beltagy, "Using deep neural networks for extracting sentiment targets in arabic tweets," in *Studies in Computational Intelligence*, vol. 740, Springer Verlag, pp. 3–15, 2018.

[38] K. H. Manguri, R. N. Ramadhan, and P. R. Mohammed Amin, "Twitter Sentiment Analysis on Worldwide COVID-19 Outbreaks," *Kurdistan J. Appl. Res.*, pp. 54–65, May 2020.

[39] R. Yadav and N. Sharma, "Analysis of Brand Value Prediction based on Social Media Data," *Int. Res. J. Eng. Technol.*, vol. 06, no. 08, pp. 1274–1278, 2019.

[40] Z. Nassr, N. Sael, and F. Benabbou, "A comparative study of sentiment analysis approaches," in *ACM International Conference Proceeding Series*, 2019.

[41] K. Dashtipour, A. Hussain, Q. Zhou, A. Gelbukh, A. Y. A. Hawalah, and E. Cambria, "PerSent: A freely available persian sentiment lexicon," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10023 LNAI, pp. 310–320, 2016.

[42] V. Agarwal, P. Aher, and V. Sawant, "Automated Aspect Extraction and Aspect Oriented Sentiment Analysis on Hotel Review Datasets," in *Proceedings - 2018 4th International Conference on Computing, Communication Control and Automation, ICCUBEA 2018*, 2018.

[43] S. M. Alzanin and A. M. Azmi, "Rumor detection in Arabic tweets using semi-supervised and unsupervised expectation–maximization," *Knowledge-Based Syst.*, vol. 185, p. 104945, Dec. 2019.

[44] N. Albadi, M. Kurdi, and S. Mishra, "Hateful people or hateful bots? Detection and characterization of bots spreading religious hatred in Arabic social media," *Proc. ACM Human-Computer Interact.*, vol. 3, no. CSCW, pp. 1–25, Nov. 2019.

[45] R. Baly, H. Hajj, N. Habash, K. B. Shaban, and W. El-Hajj, "A sentiment treebank and morphologically enriched recursive deep models for effective sentiment analysis in Arabic," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 16, no. 4, p. 23 (21 pages), Jul. 2017.

[46] R. Eskander and O. Rambow, "SLSA: A Sentiment Lexicon for Standard Arabic," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2545–2550, 2015.

[47] A. Mahmoud and T. Elghazaly, "Using twitter to monitor political sentiment for Arabic slang," in *Studies in Computational Intelligence*, vol. 740, Springer Verlag, pp. 53–66, 2018.

[48] M. A. Sghaier and M. Zrigui, "Sentiment analysis for Arabic e-commerce websites," in *Proceedings - 2016 International Conference on Engineering and MIS, ICEMIS 2016*, 2016.

[49] N. F. Bin Hathlian and A. M. Hafezs, "Sentiment - Subjective analysis framework for Arabic social media posts," in *2016 4th Saudi International Conference on Information Technology (Big Data Analysis), KACSTIT 2016*, 2016.

[50] R. Bouchlaghem, A. Elkhelifi, and R. Faiz, "Sentiment analysis in arabic twitter posts using supervised methods with combined features," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9624 LNCS, pp. 320–334, 2018.

[51] S. Al-Saqqa, N. Obeid, and A. Awajan, "Sentiment Analysis for Arabic Text using Ensemble Learning," in *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, vol. 2018-November, 2019.

[52] J. O. Atoum and M. Nouman, "Sentiment analysis of Arabic Jordanian dialect tweets," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 2, pp. 256–262, 2019.

[53] B. Haidar, M. Chamoun, and A. Serhrouchni, "Multilingual cyberbullying detection system: Detecting cyberbullying in Arabic content," in *2017 1st Cyber Security in Networking Conference, CSNet 2017*, vol. 2017-January, pp. 1–8, 2017.

[54] M. Abdullah, M. Hadzikadicy, and S. Shaikhz, "SEDAT: Sentiment and Emotion Detection in Arabic Text Using CNN-LSTM Deep Learning," in *Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018*, pp. 835–840, 2019.

[55] S. Alhumoud, T. Albuhairi, and M. Altuwaijri, "Arabic sentiment analysis using WEKA a hybrid learning approach," in *IC3K 2015 - Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, vol. 1, pp. 402–408, 2015.

[56] T. Elghazaly, A. Mahmoud, and H. A. Hefny, "Political sentiment analysis using twitter data," in *ACM International Conference Proceeding Series*, vol. 22-23-March, pp. 1–5, 2016.

[57] S. R. El-Beltagy, T. Khalil, A. Halaby, and M. Hammad, "Combining lexical features and a supervised learning approach for arabic sentiment analysis," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9624 LNCS, pp. 307–319, 2018.

[58] A. Amjad and U. Qamar, "UAMSA: Unified approach for multilingual sentiment analysis using GATE," in *ACM International Conference Proceeding Series*, pp. 1–5, 2019.

[59] M. A. Jerbi, H. Achour, and E. Souissi, "Sentiment Analysis of Code-Switched Tunisian Dialect: Exploring RNN-Based Techniques," in *Communications in Computer and Information Science*, vol. 1108, pp. 122–131, 2019.

[60] F. Amiri, S. Scerri, and M. H. Khodashahi, "Lexicon-based sentiment analysis for Persian text," in *International Conference Recent Advances in Natural Language Processing, RANLP*, vol. 2015-Jan., pp. 9–16, 2015.

BIOGRAPHIES OF AUTHORS

Youssra Zahidi is a Ph.D. student in Computer Science, Information System, and Software Engineering Laboratory, Abdelmalek Essaadi University, Tetuan, Morocco. She is a Computer Sciences engineer, graduated in 2017 from the National School of Applied Sciences, Abdelmalek Essaadi University.

Yacine El Younoussi is a Ph.D. doctor and professor of computer sciences at the National School of Applied Sciences of Tetuan, Information System and Software Engineering Laboratory, Abdelmalek Essaadi University, Tetuan, Morocco. He is a supervisor of many Thesis, and he is part of many boards of international journals and international conferences.

Yassine Al-Amrani is a Ph.D. doctor and professor of Computer Sciences at the Multidisciplinary Faculty of Larache, Abdemalek Essaadi University, Morocco. He is a Computer Sciences Engineer, and he is part of many boards of International Journals and International Conferences.