The Journal of Open Source Software

PyArabic: A Python package for Arabic text

Taha Zerrouki [1¶]

1 Bouira University, Bouira, Algeria ¶ Corresponding author

DOI: 10.21105/joss.04886

Software

- Review
- Repository
- Archive

Editor: Andrew Stewart

Reviewers:

- @amitkumarj441
- @kikarimullah

Submitted: 12 September 2022
Published: 20 April 2023

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Summary

Because text is the most common type of information representation, text processing and manipulation require recurring routines and functions. Every day, massive amounts of text are processed. Indeed, with the advent of artificial intelligence and new machine learning and deep learning enhancements, natural language processing has become a critical domain.

PyArabic is a collection of modules that provide basic functionality for manipulating Arabic texts, phrases, words, numbers, and letters. It primarily provides preprocessing tools such as normalization, tokenization, diacritics removal, number conversion, transliteration, and so on.

For years, researchers and developers who worked on machine learning algorithms for natural language processing have used the library for Arabic text preprocessing and cleaning. The library becomes more important for machine learning.

Statement of need

PyArabic is a Natural Language Processing Python package for Arabic text[1]. It is a simple library with basic functions for manipulating Arabic letters and text, such as detecting Arabic letters, Arabic letter groups and characteristics, removing diacritics, and so on. It contains the most basic and useful routines used by developers and researchers working with Arabic texts. Some key features are as follows:

- Text tokenization.
- Remove diacriticts (Harakat) from words (all, except Shadda, Tatweel, last haraka).
- Separate a word into letters and diacritics.
- Reduce diactritics of words.
- Measure tashkeel similarity (Harakats, fully or partially vocalized similarity with a template).
- Letter normalization (ligatures and Hamza).
- Numbers to words.
- Extract numerical phrases and prevocalize it.
- Unshaping texts to handle letter glyphs.
- Convert encoding and transliteration.

The PyArabic package includes five major submodules:

- Araby: Basic tools and routines for manipulating Arabic text and letters, such as tokenization and diacritics removal, are provided.
- Number: Contains routines for dealing with numbers and numeric words; allows conversion of numbers to words and words to numbers; detects numeric phrases, and more.
- Named: Provides simple tools for extracting named entities from text.

[1]The library can be found at [PyPi.org index](https://pypi.org/project/PyArabic/)

- Trans: Provides functions for converting between Arabic transliterations such as SAMPA, TIM Bukwalter, and Unicode.
- Normalize: Utility functions that are used to prepare an Arabic text for searching and indexing.
- More advanced projects use PyArabic, such as Adawat, which is an open framework for processing Arabic that the author developed as part of his PhD research. In our PhD work, we release a set of tools, the most important of which are:
  - Tashaphyne, Arabic Light Stemming Library (Zerrouki, 2022d). We primarily use tokenization, diacritics removal, and letter constants from PyArabic in this basic library.
  - Qalsadi is an Arabic morphology analyzer (Zerrouki, 2022b), which is based on Tashaphyne Stemmer, It uses Pyarabic, especially for tokenization and letters and diacritics handling, which includes removing tashkeel, handling Shadda, removing the last diacritic for inflection cases, and comparing two words with full or partial diacritization.
  - Qutrub (Zerrouki, 2022c) is an Arabic verb conjugator, and this conjugation library requires basic features such as the separation of diacritics from letters and rejoining them to form words during conjugation. normalizing letters and words to prepare them for conjugation.
  - Mishkal, is a system for Arabic text diacritization (Zerrouki, 2022a). It is built on cited libraries like the Qalsadi morphology analyzer, the Tashaphyne stemmer, the Qutrub conjugator, and others. For basic routines, it uses PyArabic for letter constant names, diacritics management, word normalization, tokenization, and numeric phrase detection.

  The Classical Language Toolkit (CLTK)[2] (Johnson, 2014) provides natural language processing support for Ancient, Classical, and Medieval Eurasia languages. CLTK integrates PyArabic functionalities for corpus importer, tokenization, text converting, and transliteration for classical Arabic (Johnson, 2014), which is the form of the Arabic language used in texts from the 7th century AD to the 9th century AD (like the orthography of the Quran).

PyArabic was created to aid researchers and developers in natural language processing tasks, particularly text preprocessing (tokenization, cleaning, normalization, strip diacritics). It has already appeared in several scientific publications. It is mentioned in:

- Text alignment (Mikhael, 2014).
- Text classification (Abozinadah & Jones Jr, 2016; Abufayad, 2018; Ajlouni, 2021; AlBatayha, 2021; Habash, 2021; Mgheed, 2021).
- Sentiment analysis (Al-Hagery et al., 2020; Alharbi et al., 2020; Al-Horaibi & Khan, 2016; Almutairi & Al-Hagery, 2021; Alotaibi et al., 2019; Kaibi et al., 2019, 2020; Khabour et al., 2022; Mihi, Ali, et al., 2020; Mihi, Ait, et al., 2020; Mihi et al., 2022; Oussous et al., 2020).
- Language model (Alzu'bi & Duwairi, 2021; Hamed et al., 2017).
- Text preprocessing (remove diacritics, tokenization, etc.): Zhang et al. (2021)
- Lexical resources (Choe et al., 2020)
- Text similarity (Mouty & Gazdar, 2019)

PyArabic was inspired by Ar-PHP(Al-Shamaa, 2022), an Arabic library for the PHP programming language that provides basic routines for web developers. Then the two libraries grow together through collaborations, and they are inspired mutually by each other. Ar-PHP provides basic routines for PHP and MySQL databases and attempts to solve web development issues such as arabic glyph rendering; however, the Ar-PHP library also includes advanced modules such as sentiment analysis, muslim prayer times, and auto-summarize (Al-Shamaa, 2022).

[2] http://cltk.org

There are many dedicated frameworks for Arabic natural language processing, like MADAMIRA(Java) (Pasha et al., 2014), FARASA(Java)(Abdelali et al., 2016), CAMeL(Python) (Obeid et al., 2020). Many multilingual frameworks, however, such as NLTK (Python) (Loper & Bird, 2002), Spacy (Python) (Vasiliev, 2020), and CLTK (Python) (Johnson, 2014), only partially support Arabic.

In PyArabic, we focused on basic routines and build our library to be native and independent enough to be embedded in complex projects. This library was used in many projects and adopted by frameworks like CLTK(Johnson, 2014), and has been inspired to build more specific libraries like TaKseem (a tokenization library for Arabic) (Alyafeai & Saeed, 2020b) and Tankeeh (Arabic cleaning, normalization, and segmentation library) (Alyafeai & Saeed, 2020a).

Acknowledgements

We gratefully acknowledge the contributions of Assem Chelli, Khaled Alshamaa, Lakhdar Benzahia, Mouhamad AboShokor, David Lowe, Ahmed Alq, and Arabeyes.org during the project's inception.

References

Abdelali, A., Darwish, K., Durrani, N., & Mubarak, H. (2016). Farasa: A fast and furious segmenter for arabic. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 11–16. https://doi.org/10.18653/v1/N16-3003

Abozinadah, E. A., & Jones Jr, J. H. (2016). Improved microblog classification for detecting abusive arabic twitter accounts. *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, *6*(6), 17–28. https://doi.org/10.5121/ijdkp.2016.6602

Abufayad, T. I. (2018). *Semantic word clustering from large arabic text* [PhD thesis]. The Islamic University of Gaza.

Ajlouni, M. (2021). Experience simple transformer library in solving mojaz multi-topic labelling task. *2021 12th International Conference on Information and Communication Systems (ICICS)*, 466–467. https://doi.org/10.1109/icics52457.2021.9464602

Alasmari, A., Alhothali, A., & Allinjawi, A. (2022). Hybrid machine learning approach for arabic medical web page credibility assessment. *Health Informatics Journal*, *28*(1), 14604582211070998. https://doi.org/10.1177/14604582211070998

AlBatayha, D. (2021). Multi-topic labelling classification based on LSTM. *2021 12th International Conference on Information and Communication Systems (ICICS)*, 471–474. https://doi.org/10.1109/ICICS52457.2021.9464531

Al-Hagery, M. A., Al-Assaf, M. A., & Al-Kharboush, F. M. (2020). Exploration of the best performance method of emotions classification for arabic tweets. *Indonesian Journal of Electrical Engineering and Computer Science*, *19*(2), 1010–1020. https://doi.org/10.11591/ijeecs.v19.i2.pp1010-1020

Alharbi, B., Alamro, H., Alshehri, M., Khayyat, Z., Kalkatawi, M., Jaber, I. I., & Zhang, X. (2020). Asad: A twitter-based benchmark arabic sentiment analysis dataset. *KAUST Arabic Sentiment Analysis Challenge*.

Al-Horaibi, L., & Khan, M. B. (2016). Sentiment analysis of arabic tweets using text mining techniques. *First International Workshop on Pattern Recognition*, *10011*, 288–292. https://doi.org/10.1117/12.2242187

Al-Jamaan, R., Ykhlef, M., & Alothaim, A. (2022). FluSa-tweet: A benchmark dataset for influenza detection in saudi arabia. *2022 13th International Conference on Information and Communication Systems (ICICS)*, 346–351. https://doi.org/10.1109/icics55353.2022.9811149

Almutairi, A. R., & Al-Hagery, M. A. (2021). Cyberbullying detection by sentiment analysis of tweets' contents written in arabic in saudi arabia society. *International Journal of Computer Science & Network Security*, *21*(3), 112–119.

Alotaibi, S., Mehmood, R., & Katib, I. (2019). Sentiment analysis of arabic tweets in smart cities: A review of saudi dialect. *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*, 330–335. https://doi.org/10.1109/fmec.2019.8795331

Alrumayyan, N., & Al-Yahya, M. (2022). Neural embeddings for the elicitation of jurisprudence principles: The case of arabic legal texts. *Applied Sciences*, *12*(9), 4188. https://doi.org/10.3390/app12094188

Al-Sarem, M., Alsaeedi, A., & Saeed, F. (2020). A deep learning-based artificial neural network method for instance-based arabic language authorship attribution. *International Journal of Advances in Soft Computing and Its Applications*, *12*(2).

Al-Shamaa, K. (2022). *Ar-PHP, PHP library for website developers to process arabic content* (Version 6.3.1). https://github.com/khaled-alshamaa/ar-php

Alyafeai, Z., & Saeed, M. (2020a). Tkseem: A preprocessing library for arabic. In *GitHub repository*. https://github.com/ARBML/tnkeeh; GitHub.

Alyafeai, Z., & Saeed, M. (2020b). Tkseem: A tokenization library for arabic. In *GitHub repository*. https://github.com/ARBML/tkseem; GitHub.

Alzu'bi, D., & Duwairi, R. (2021). Detecting regional arabic dialect based on recurrent neural network. *2021 12th International Conference on Information and Communication Systems (ICICS)*, 90–93. https://doi.org/10.1109/icics52457.2021.9464605

Choe, Y. J., Park, K., & Kim, D. (2020). word2word: A collection of bilingual lexicons for 3,564 language pairs. *Proceedings of the 12th Language Resources and Evaluation Conference*, 3036–3045.

Duwairi, R., Hayajneh, A., & Quwaider, M. (2021). A deep learning framework for automatic detection of hate speech embedded in arabic tweets. *Arabian Journal for Science and Engineering*, *46*(4), 4001–4014. https://doi.org/10.1007/s13369-021-05383-3

Elouali, A., Elberrichi, Z., & Elouali, N. (2020). Hate speech detection on multilingual twitter using convolutional neural networks. *Revue d'Intelligence Artificielle*, *34*(1), 81–88. https://doi.org/10.18280/ria.340111

Habash, M. (2021). Team MohammadHabash at mowjaz multi-topic labelling task. *2021 12th International Conference on Information and Communication Systems (ICICS)*, 468–470. https://doi.org/10.1109/ICICS52457.2021.9464614

Hamed, I., Elmahdy, M., & Abdennadher, S. (2017). Building a first language model for code-switch arabic-english. *Procedia Computer Science*, *117*, 208–216. https://doi.org/10.1016/j.procs.2017.10.111

Johnson, K. (2014). *CLTK: The classical language toolkit*. https://github.com/cltk/cltk.

Kaibi, I., Nfaoui, E. H., & Satori, H. (2019). A comparative evaluation of word embeddings techniques for twitter sentiment analysis. *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, 1–4. https://doi.org/10.1109/wits.2019.8723864

Kaibi, I., Nfaoui, E. H., & Satori, H. (2020). Sentiment analysis approach based on combination of word embedding techniques. In *Embedded systems and artificial intelligence* (pp. 805–813). Springer. https://doi.org/10.1007/978-981-15-0947-6_76

Khabour, S. M., Al-Radaideh, Q. A., & Mustafa, D. (2022). A new ontology-based method for arabic sentiment analysis. *Big Data and Cognitive Computing*, *6*(2), 48. https://doi.org/10.3390/bdcc6020048

Loper, E., & Bird, S. (2002). Nltk: The natural language toolkit. *arXiv Preprint Cs/0205028*.

Marie-Sainte, S. L. (2022). Samee'a: A new system for arabic recitation using speech recognition and jaro winkler algorithm: Samee'a arabic recitation. *Kuwait Journal of Science*, *49*(1).

Mgheed, R. M. A. (2021). Scalable arabic text classification using machine learning model. *2021 12th International Conference on Information and Communication Systems (ICICS)*, 483–485. https://doi.org/10.1109/icics52457.2021.9464566

Mihi, S., Ait, B., El, I., Arezki, S., & Laachfoubi, N. (2020). MSTD: Moroccan sentiment twitter dataset. *International Journal of Advanced Computer Science and Applications*, *11*(10), 363–372. https://doi.org/10.14569/ijacsa.2020.0111045

Mihi, S., Ali, B. A. B., Bazi, I. E., Arezki, S., Laachfoubi, editor="Serrhini., Nabil", Silva, C., & Aljahdali, S. (2020). A comparative study of feature selection methods for informal arabic. *Innovation in Information Systems and Technologies to Support Learning Research*, 203–213. https://doi.org/10.1007/978-3-030-36778-7_22

Mihi, S., Ali, B. A. B., El Bazi, I., Arezki, S., & Laachfoubi, N. (2022). Dialectal arabic sentiment analysis based on tree-based pipeline optimization tool. *International Journal of Electrical and Computer Engineering (IJECE)*, *12*(4), 4195–4205. https://doi.org/10.11591/ijece.v12i4.pp4195-4205

Mikhael, K. A. (2014). The greek-arabic new testament interlinear process: Greekarabicnt. org. *LRE-REL2*, 1.

Mouty, R., & Gazdar, A. (2019). The effect of the similarity between the two names of twitter users on the credibility of their publications. *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, 196–201. https://doi.org/10.1109/iciev.2019.8858561

Nguyen, K., & Daumé, H. (2019). *Global voices: Crossing borders in automatic news summarization*. arXiv. https://doi.org/10.48550/ARXIV.1910.00421

Obeid, O., Zalmout, N., Khalifa, S., Taji, D., Oudah, M., Alhafni, B., Inoue, G., Eryani, F., Erdmann, A., & Habash, N. (2020). CAMeL tools: An open source python toolkit for Arabic natural language processing. *Proceedings of the 12th Language Resources and Evaluation Conference*, 7022–7032. ISBN: 979-10-95546-34-4

Oussous, A., Benjelloun, F.-Z., Lahcen, A. A., & Belfkih, S. (2020). ASA: A framework for arabic sentiment analysis. *Journal of Information Science*, *46*(4), 544–559. https://doi.org/10.1177/0165551519849516

Pasha, A., Al-Badrashiny, M., Diab, M., 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. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 1094–1101.

Solyman, A., Wang, Z., Tao, Q., Elhag, A. A. M., Zhang, R., & Mahmoud, Z. (2022). Automatic arabic grammatical error correction based on expectation-maximization routing and target-bidirectional agreement. *Knowledge-Based Systems*, *241*, 108180. https://doi.org/10.1016/j.knosys.2022.108180

Sun, J., Ahn, H., Park, C., Tsvetkov, Y., & Mortensen, D. (2021). Cross-cultural similarity features for cross-lingual transfer learning of pragmatically motivated tasks. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, 2403–2414. https://doi.org/10.18653/v1/2021.eacl-main.204

Taha, M., & Barakat, N. (2022). Arabic image captioning: The effect of text pre- processing on the attention weights and the BLEU-n scores. *International Journal of Advanced Computer Science and Applications*, *13*, 2022. https://doi.org/10.14569/IJACSA.2022.0130751

Tarmom, T., Atwell, E., & Alsalka, M. (2019). Non-authentic hadith corpus: Design and methodology. *International Conference on Islamic Applications in Computer Science and Technologies (IMAN 2019)*.

Vasiliev, Y. (2020). *Natural language processing with python and SpaCy: A practical introduction*. No Starch Press.

Yusuf, N., Mohd Yunus, M. A., & Wahid, N. (2019). Arabic text stemming using query expansion method. *International Conference of Reliable Information and Communication Technology*, 3–11. https://doi.org/10.1007/978-3-030-33582-3_1

Zerrouki, T. (2022a). Mishkal arabic text vocalization software. In *GitHub repository*. GitHub. https://github.com/linuxscout/mishkal

Zerrouki, T. (2022b). Qalsadi arabic morphological analyzer and lemmatizer for python. In *GitHub repository*. GitHub. https://github.com/linuxscout/qalsadi

Zerrouki, T. (2022c). Qutrub: Arabic verb conjugation software. In *GitHub repository*. GitHub. https://github.com/linuxscout/qutrub

Zerrouki, T. (2022d). Tashaphyne: Arabic light stemmer. In *GitHub repository*. GitHub. https://github.com/linuxscout/tashaphyne

Zhang, X., Yang, Q., Albaradei, S., Lyu, X., Alamro, H., Salhi, A., Ma, C., Alshehri, M., Jaber, I. I., Tifratene, F., & others. (2021). Rise and fall of the global conversation and shifting sentiments during the COVID-19 pandemic. *Humanities and Social Sciences Communications*, *8*(1), 1–10. https://doi.org/10.1057/s41599-021-00798-7