Webology (ISSN: 1735-188X)
Volume 19, Number 5, 2022

A Pyarabic Python Library To Create Arabic Applications

kawakib Mahmood Hussien [1], Nadia Mahmood Hussien [2], Yasmin Makki Mohialden [3]

[1]University of Baghdad - College of Education Ibn Rushd, Baghdad-Iraq.

[2,3]Computer Science Department, Collage of Science, Mustansiriyah University, Baghdad-Iraq.

Abstract

The Python programming language offers a smart package called PyArabic, which is a set of tools that help with the Arabic language. It sorts letters into groups, sorts text into sentences or words, deletes animations, separates and combines movements in texts, reduces changes, measures symmetry between two words, profiles characters, and pulls numerical sentences from syntax. raw text for numeric expressions and reversed Arabic script for systems that don't support character grids. This paper describes what this package can do and how it can be used in many systems. In this research paper, we talk about the most important things that can be done with this library right now.

Keywords Python, PyArabic, Python-based smart apps, Arabic language programming.

1. Introduction

Pyarabic A Python library for the Arabic language contains functions for finding Arabic letters, Arabic letter groups, and characteristics, erasing diacritics, and performing other core operations. The Tashaphyne Library is also included for word normalization. Pyarabic is a comparable software with the same goal. It handles the Arabic language's nuances and makes it easier to absorb a variety of Arabic texts. The obtained Arabic documents were vectorizable throughout the preprocessing procedure. The corpus has to be standardized and broken apart as part of the preparatory stages required for the experiments and system installation. During the normalization process, special characters and diacritical markings were removed from the data. To normalize the text, the PyArabic Python module was utilized [1, 2, 3]. The best way to deal with Arabic text in Python is to use Unicode encoding, which is built into Python and does not require any additional libraries or functions. This is arguably the most important reason we chose Python: all you have to do is enter the letter U before the content, and Python will handle it transparently [1]. Taha Zerrouki built this package, which may be found at http://tahadz.com. The programs in the package are for processing Arabic text, and the advantages of PyArabic include the ability to arrange letters by type and text into units such as sentences or words. Delete all animations except intensity (intensity, lengthening, and the last movement); Modulation can be reduced by separating and integrating movements from sentences. Determine the symmetry of two words (in partial and total movements, as well as weight symmetry); Character analysis (standardization of compositions such as "lam" and "whispers"), and numbers to words conversion Look for numerical phrases in the text; create numerical statements for the first time. Flip Arabic text for systems that do not support character networking.

PyArabic can normalize letters (ligatures and hamza), split texts into words or sentences, separate and join letters and harakat, reduce tashkeel, measure tashkeel similarity (harakat, fully or partially vocalized, similarity with a template), vocalize numerical phrases in advance, extract numerical phrases, and unship texts using groups of Arabic letters.
The design for this study is divided into five sections: two regarding related work and three conclusions.

2. Related Work

Here are some examples of works created with the PyArabic Package and Python: - Shawar, Abu, and E. S. Atwell's 2014 book discusses how machine-learning methods were used to create an Arabic chatbot that users may communicate to in Arabic and get answers from the Qur'an. A system that learns conversational patterns from a corpus of transcribed conversations was used to create a variety of chatbots that speak a variety of languages, including English, French, and Afrikaans. We explain the improved process for dealing with Arabic training material and input/output, as well as characteristics of the Arabic language that cause issues for chatbot learning. They used the Qur'an as a training corpus for our chatbot since it generally offers guidance and answers to concerns about religion and other topics. They altered the learning method to account for the Qur'an's structure in terms of sooras and ayyats because it is not a record of a discussion. As a result, users can type in Arabic and receive responses from the Qur'an.
In 2016, [Al Hagbani, Eman Saad, and Muhammad Badruddin Khan] demonstrated BOTTA, the first Arabic-speaking Chabot. We discuss the difficulties of creating a conversational AI that attempts to imitate amicable talks in the Egyptian-Arabic dialect. We go over alternative solutions and break down the different parts of the BOTTA Chabot. Researchers working on Arabic Chabot technology have access to the BOTTA database files. Anyone who wants to chat with the BOTTA Chabot online can do so [5].

In the same year, Ehab A. Abozinadah came up with a way to stop spammers from using Arabic Twitter accounts to send adult-oriented messages in Arabic. He called it "Arabic Word Correction." This kind of material isn't allowed in Arabic-speaking countries because it goes against their culture. With 96.5% accuracy, we can tell from our method which Arabic Twitter accounts are abusive [6].

Muhammad Badruddin Khan and Lamia Al-Horaibi used the Decision Tree algorithm, machine learning, and the Naive Bayes algorithm to make an early model that accurately measures how Arabic Twitter users feel as of 2016. Approximately 2,000 Arabic Tweets were taken from Twitter and put into the datasets that were used. To see how well the two algorithm classifiers worked, we did a number of tests with diverse groupings of text-processing functions. We discovered that the tools for processing Arabic text should be created from scratch or made better if we want to make accurate classifiers. The small features we made in Python were crucial in improving the results and showed that the vocabulary for sentiment analysis in the Arabic domain requires a lot of development. [7]. In 2017, Abozinadah, Ehab A, and others wanted to find offensive accounts that upload pornographic information in Arabic so that Arab utterers could see it. There aren't many natural language processing (NLP) tools for the Arabic language, and with the data that we have, there hasn't been any research to find social media accounts for adults that use Arabic. We looked at the content of Twitter using statistical learning to find abusive accounts that utilize slang, vulgarity, obscenities, and swear words in Arabic. Our method for predicting the future was 96% accurate, and it did better than copies of the bag-of-words (BOW) method [8].

In 2019, [Kaibi, Ibrahim, and others] gave a comparison of Twitter sentiment analysis word embedding approaches.

The classification outcome is dependent on the representation of the text and the retrieved characteristics used to train the classifier. For many text mining tasks, such as sentiment analysis, word embeddings have been shown to be a useful technique for generating word representations. In this work, we investigate the Word2Vec, FastText, and Glove word embedding methods using Twitter datasets for sentiment analysis. This is accomplished by using six well-known machine learning techniques: GaussianNB, LinearSVC, NuSVC, Logistic Regression, SGD, and RandomForest. FastText representation with NuSVC, a sort of SVM classifier, performs better in terms of accuracy than the other combinations [9].

In [2020], Kaibi, Ibrahim, and others demonstrate that sentiment analysis depends on word embeddings that have already been trained, notably the Arabic and FastText models. We offered a combination of these two models based on the vector concatenation of the two models. Six different machine learning techniques were used to classify sentiment. Most of the time, the most accurate results come from our suggested method, especially when it is combined with a NuSVC classifier, which is a type of SVM [10].

Mohammad Habash, in the year 2021, This paper explains how to classify articles via a bidirectional gated recurrent unit (Bi-GRU) with AraVec embeddings. We wish to determine what type of article it is based on its contents. Mowjaz articles are included in the dataset used for this investigation. Mowjaz is a smartphone application that allows users to stay current on news, sports, entertainment, and other topics from famous Arabic media. The F1 score for our system is 0.8344, which is much higher than the baseline models [11].

Moataz examines the code used in Ajlouni's [2021] description of a multitopic tagging system. After training, validation, and test data sets have been loaded; the code installs the PyArabic and Simple Transformers libraries. "Pyarabic" gives the program the ability to handle Arabic letters. The "Simple Transformers" Natural Language Processing (NLP) package makes it simpler to implement Transformer models without compromising their efficacy. The Simple Transformer library was used to obtain the "Multi-Label Classification Model" with the model type "bert" and model name "asafaya/bert-base-arabic". The target result of multi-label text classification is a list of ten distinct binary labels for each article (row) in the training dataset. Typically, a transformer-based multi-label text classification technique comprises a transformer model atop a classification layer. The results were favorable despite the fact that training for two epochs required only 5 minutes and 27 seconds. F1 macro had an accuracy of 0.866, F1 micro had an accuracy of 0.866, and the competition webpage on Codalab had an accuracy of 0.8468 [12].

In the same year, Rahaf M. Al-Mgheed utilized an SVM classifier to develop a model for categorizing Arabic text with multiple labels. This model is predominantly utilized in the classification of articles according to their subject matter. Utilizing the SVM classifier on the dataset yielded the most accurate outcomes with an accuracy of 82.2%. The model was crafted using Python.

3. Conclusion

In this paper, we look at PyArbic, a Python library for the Arabic language. PyArbic lets you do basic things with Arabic texts and letters, such as finding Arabic letters, Arabic letter characteristics, and Arabic letter groups and removing diacritics. We also take a look at the most important applications and books that can be used with this library to make a useful Arabic-language cross-platform app.

References

[1] T. Zerrouki, “PyArabic: Arabic text tools for Python.” Accessed: Sep. 13, 2022. [OS Independent]. Available: http://pyarabic.sourceforge.net/

[2] A. Oussous, F.-Z. Benjelloun, A. Ait Lahcen, and S. Belfkih, “ASA: A framework for Arabic sentiment analysis,” J. Inf. Sci., vol. 46, p. 016555151984951, May 2019, doi: 10.1177/0165551519849516.

[3] Alswedani, S.; Katib, I.; Abozinadah, E.; Mehmood, R. Discovering Urban Governance Parameters for Online Learning in Saudi Arabia During COVID-19 Using Topic Modeling of Twitter Data. Front. Sustain. Cities 2022, 4, 1–24, doi:10.3389/FRSC.2022.751681

[4]B. A. Abu Shawar and E. S. Atwell, "An Arabic chatbot giving answers from the Qur'an," Proceedings of TALN04: XI Conference sur le Traitement Automatique des Langues Naturelles, 2004. https://eprints.whiterose.ac.uk/82455/ (accessed Sep. 14, 2022).

[5]E. S. AlHagbani and M. B. Khan, "Challenges facing the development of the Arabic chatbot," in First International Workshop on Pattern Recognition, Jul. 2016, vol. 10011, pp. 192–199. doi: 10.1117/12.2240849.

[6]E. A. Abozinadah, "Improved Micro-Blog Classification for Detecting Abusive Arabic Twitter Accounts." Rochester, NY, 2016. Accessed: Sep. 14, 2022. [Online]. Available: https://papers.ssrn.com/abstract=3628999

[7]L. Al-Horaibi and M. B. Khan, "Sentiment analysis of Arabic tweets using text mining techniques," in First International Workshop on Pattern Recognition, Jul. 2016, vol. 10011, pp. 288–292. doi: 10.1117/12.2242187.

[8] Abozinadah, Ehab A., and James H. Jones Jr. "A statistical learning approach to detect abusive twitter accounts." Proceedings of the International Conference on Compute and Data Analysis. 2017.

[9I Kaibi, E. H. Nfaoui, and H. Satori, "A Comparative Evaluation of Word Embeddings Techniques for Twitter Sentiment Analysis," in 2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS), Apr. 2019, pp. 1–4. doi: 10.1109/WITS.2019.8723864.

[10] Kaibi, Ibrahim, El Habib Nfaoui, and Hassan Satori. "Sentiment analysis approach based on combination of word embedding techniques." Embedded Systems and Artificial Intelligence. Springer, Singapore, 2020. 805-813.

[11] Habash, Mohammad. "Team MohammadHabash at Mowjaz Multi-Topic Labelling Task." 2021 12th International Conference on Information and Communication Systems (ICICS). IEEE, 2021.

[12]Ajlouni, Moataz. "Experience Simple Transformer library in solving Mojaz Multi-Topic Labelling Task." 2021 12th International Conference on Information and Communication Systems (ICICS). IEEE, 2021.

[13]Mgheed, Rahaf M. AL. "Scalable arabic text classification using machine learning model." 2021 12th International Conference on Information and Communication Systems (ICICS). IEEE, 2021.