Morphkit Documentation

Morphkit logo

# Home[](#home)

Welcome to the documentation for Morphkit, a Python toolkit for processing the output of the Morpheus Morphological analyzer.

This package was created as part of a research project to create a [Text-Fabric dataset](https://tonyjurg.github.io/N1904addons/) containing the Morpheus analytical data for each word of the [Nestle1904](https://centerblc.github.io/N1904/) Greek New Testament. A number of functions are specificly related to this use case.

# Features[](#features)

* Lightweight and modular morphological toolkit.
* Compatible with Morpheus environments.
* Designed for use with Greek New Testament texts (SP tags).
* Basic support for Latin.

# Using this package[](#using-this-package)

[Installation](#document-install)

How to install this package in your Python environments

[Usage](#document-usage)

How to use this package

# GitHub[](#github)

You can find the project’s source code on [GitHub](https://github.com/tonyjurg/morphkit) and report issues or suggestions at the [issue tracker](https://github.com/tonyjurg/morphkit/issues).

## Installation[](#installation)

Since this not a ‘normal package’ (yet), you need to import it by adding the relative path to the location where the **directory** /morphkit can be found to the path-search (using sys.path.insert). After this, your notebook will be able to load morphkit.

import sys  
sys.path.insert(0, "../../morphkit") # relative to notebook dir  
import morphkit

### Dependencies[](#dependencies)

You also need to have install the [dependencies](https://github.com/tonyjurg/morphkit/blob/main/morphkit/requirements.txt):

* [beta-code-py](https://pypi.org/project/beta-code-py/) (for Beta Code ↔ Unicode conversion)
* [requests](https://pypi.org/project/requests/) (for API calls using HTTP to Morpheus)

### Morpheus[](#morpheus)

Since the purpose of this package it to interface with an Morpheus API endpoint, you need access to such a service. A Morpheus API endpoint can be set up locally, e.g., via the [perseidsproject/morpheus-api Docker image](https://hub.docker.com/r/perseidsproject/morpheus-api).

## Usage[](#usage)

Some example use cases.

### Obtain Morpheus Analytic blocks for a Greek word[](#Xa4b690b355715b624cdfad458c710a276fc3e71)

# convert unicode greek to betacode  
import beta\_code  
bc\_word=beta\_code.greek\_to\_beta\_code(u'του')  
api\_endpoint="10.0.1.156:1315"  
print(morphkit.get\_word\_blocks(bc\_word,api\_endpoint))

This will output three Morpheus Analytic blocks (one shown):

:raw tou  
  
:workw tou=  
:lem o(  
:prvb  
:aug1  
:stem tou= indeclform  
:suff  
:end masc/neut gen sg indeclform article

### Get the compact analysis results[](#get-the-compact-analysis-results)

print(morphkit.get\_word\_blocks(bc\_word,api\_endpoint,output="compact"))

This prints (for the same word as the previous example) the compact notation:

tou  
<NL>N tou=,o( masc/neut gen sg indeclform article</NL><NL>N tou=,ti/s gen sg attic indeclform indecl</NL><NL>N tis gen sg attic enclitic indeclform indef</NL>

### Perform a full analysis[](#perform-a-full-analysis)

The following will analyze a word and produce a full dictionary of Morpheus data augmented with Part of Speech and morph tag information.

result=morphkit.analyze\_word\_with\_morpheus("mo/non",api\_endpoint)

This provides a Python dictionary like below (truncated):

{'raw\_bc': 'mo/non',  
 'raw\_uc': 'μόνον',  
 'blocks': 2,  
 'analyses': [{'raw\_bc': 'mo/non',  
 'raw\_uc': 'μόνον',  
 'workw\_bc': 'mo/non',  
 'workw\_uc': 'μόνον',  
 'lem\_full\_bc': 'mo/nos',  
 'lem\_full\_uc': 'μόνος',  
 'lem\_base\_bc': 'mo/nos',  
 'lem\_base\_uc': 'μόνος',  
 'stem\_bc': 'mon',  
 'stem\_uc': 'μον',  
 'stem\_codes': ['os\_h\_on'],  
 'end\_bc': 'on',  
 'end\_uc': 'ον',  
 'gender': 'masc',  
 'case': 'acc',  
 'number': 'sg',  
 'end\_codes': ['os\_h\_on'],  
 'pos': 'noun',  
 'morph': 'N-ASM'},  
 {'raw\_bc': 'mo/non',  
 'raw\_uc': 'μόνον',  
 'workw\_bc': 'mo/non',  
 ...

### Limited Latin support[](#limited-latin-support)

Latin words can be analyzed and the results stored in a Python dictionary:

import pprint as pp  
raw\_text=morphkit.get\_word\_blocks("dico",api\_endpoint,language="latin")  
blocks=morphkit.split\_into\_raw\_blocks(raw\_text)  
all\_parses = []  
for block in blocks:  
 raw\_beta, parses = morphkit.parse\_word\_block(block,"latin")  
 all\_parses.append(parses)  
 pp.pprint(parses)

This procudes a dictiorary like:

[{'end': 'o\_',  
 'end\_codes': ['conj1'],  
 'lem\_base': 'dico#',  
 'lem\_full': 'dico#1',  
 'lem\_homonym': 1,  
 'mood': 'indicative',  
 'number': 'sg',  
 'person': '1',  
 'raw': 'dico',  
 'stem': 'dic',  
 'stem\_codes': ['conj1', 'are\_vb'],  
 'tense': 'present',  
 'voice': 'active',  
 'workw': 'dico\_'}]  
[{'end': 'o\_',  
 'end\_codes': ['conj3'],  
 'lem\_base': 'dico#',

To see these examples in action, you can download [this Jupyter Notebook](https://github.com/tonyjurg/morphkit/blob/main/notebooks/Morphkit_usage_examples.ipynb).

# Summary of functions[](#summary-of-functions)

|  |  |
| --- | --- |
| [morphkit.analyze\_pos](#X519b8919b632a78bc7d55abe260cb135f3a23d1) | Analyze a single Morpheus parse record and determine its part of speech. |
| [morphkit.analyze\_morph\_tag](#Xf29ab71b0b555222e3bcd21d466718dd3c2c5e1) | Compute the Sandborg–Petersen morphological tag for a single Morpheus analyses block. |
| [morphkit.analyze\_word\_with\_morpheus](#X7e0956514433321efbfe461e2e3c5649a1197d7) | Query the Morpheus morphological analyzer for a Greek word in Betacode and parse its analyses. |
| [morphkit.annotate\_and\_sort\_analyses](#X15ceec3e499da7a72342bc1f17d551dd1355667) | Annotate and sort analyses in a morphkit-compatible structure, grouping by base lemma and appending homonym suffixes extracted from lem\_full\_bc minus lem\_base\_bc. |
| [morphkit.compare\_tags](#Xa0cf932ce229fe9e5d28b46b4954c7b457b0099) | Compare two morphological parsing tags by decoding them into features and computing a weighted similarity score. |
| [morphkit.decode\_tag](#Xa5429a6d1ac7710dae9db5317403039f1d63219) | Decode a morphological tag into a set of human-readable features. |
| [morphkit.get\_word\_blocks](#X02a3fb7d2456a06423c4d0e66d6a15e1e57d970) | Retrieve the raw word blocks data for a given beta-code word from a Morpheus endpoint. |
| [morphkit.init\_compare\_tags](#X1f0ba9073af33000778729b950f50d628e6ffbb) | Factory that initializes and returns a fully-configured [compare\_tags()](#Xa0cf932ce229fe9e5d28b46b4954c7b457b0099) function. |
| [morphkit.parse\_word\_block](#Xeb950adac7d7970c2447db46bf6f92cbbfc0b47) | Parse a single Morpheus output block of Beta-code lines into structured morphological data. |
| [morphkit.split\_into\_raw\_blocks](#X8b1d76d632b17bee3b8ae52b918974e6fd5c086) | Split the input text into blocks at each ':raw' header using multiline regex. |