**Gnfinder:**

Very fast finder of scientific names. It uses dictionary and NLP approaches. On modern multiprocessor laptops it is able to process 15 million pages per hour. Works with many file formats and includes name verification against many biological databases. For full functionality it requires an Internet connection.

Installed as a command line app in windows, with commands:

mkdir C:\bin

copy path\_to\gnfinder.exe [C:\bin](http://../../../../bin)  
last step: adding [C:\bin](http://../../../../bin) directory to PATH environment variable.

When you run gnfinder command for the first time, it will create a [gnfinder.yml](https://github.com/gnames/gnfinder/blob/master/gnfinder/cmd/gnfinder.yml) configuration file.

Command: gnfinder test.txt -f tsv runs gnfinder for test.txt with a tsv output.

Starting as a web-application and an API server on port 8080:  
gnfinder -p 8080

[**A test**

**Input**:

Abstract

Bleheratherina pierucciae is described from Tontouta

(26°56.9’S 166°14’E) and Pirogues Rivers, New Caledonia. The new species has been compared with other IndoPacific atherinids, both freshwater and marine (representatives of genera Atherinason, Atherinomorus, Atherinosoma,

Atherion, Craterocephalus, Hypoatherina, Kestratherina,

Leptatherina and Stenatherina) and an atherionid (Atherion). Dyer & Chernoff’s (1996) division of Atherinidae

into three subfamilies has been briefly reviewed and a

fourth subfamily, Bleheratherininae, is now added to this

list since the new species is distinct and different from all

known atherinids. Bleheratherina pierucciae can be immediately recognised by the unusual structure of its mouthparts. Other distinct osteological characters confirm that it

merits a subfamilial status. The evolutionary history of this

new species must have commonality with the Australian

coastal and marine fishes, having probably been derived

from a common ancestor likely to have occurred in a

marine environment i.e. Arafura Sea. The zoogeographic

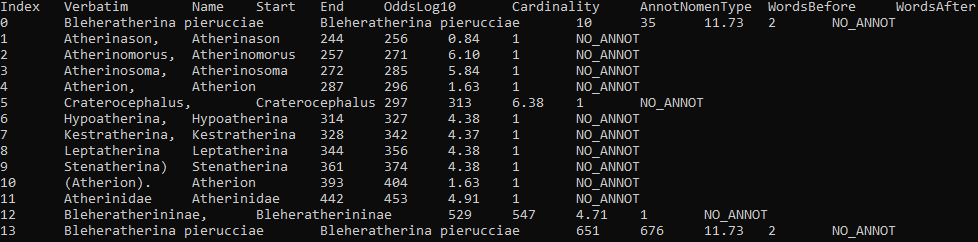
events, which led to the separation of New Caledonia from

Australia and its emergence as a separate island, post

Palaeocene, must have led to a divergence of the ancestral

fauna which invaded the freshwaters of New Caledonia.

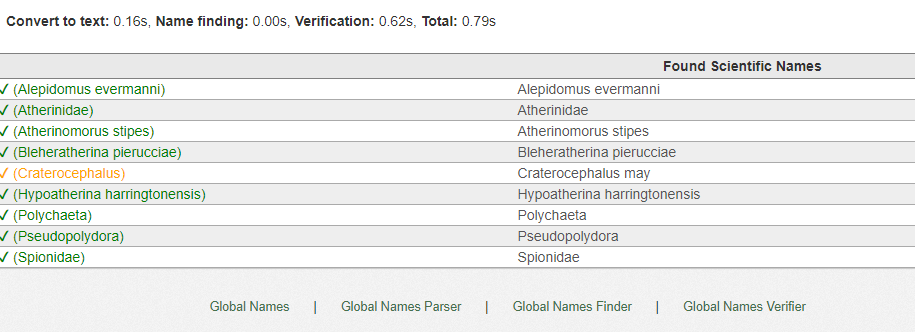
**Output:**

(the online public gnfinder didn’t catch “(Atherion).” )

]

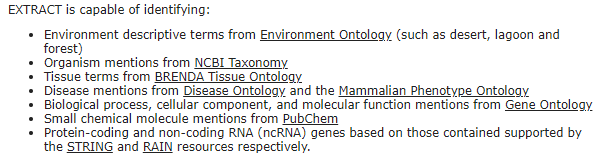
It was also tested for url and pdf.

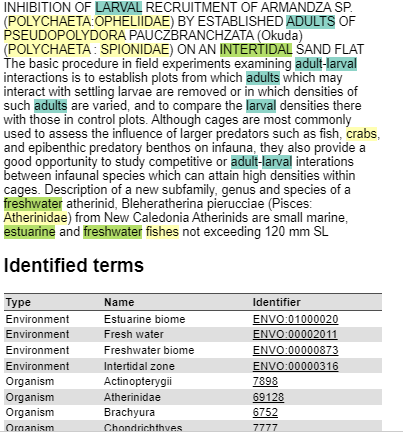
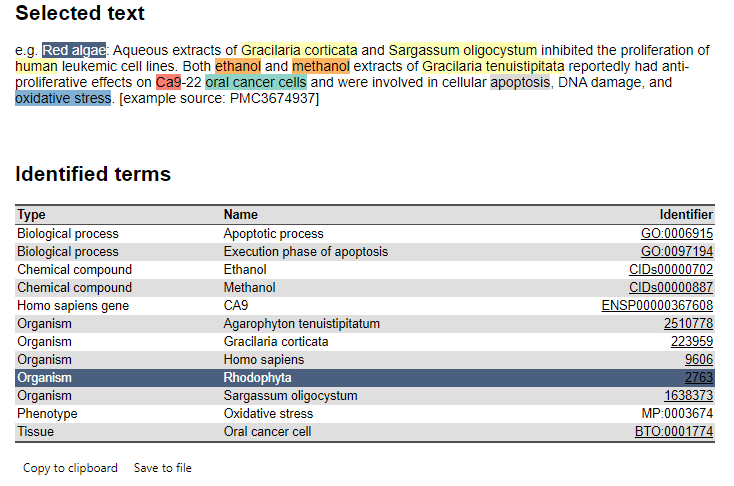
Verification example:

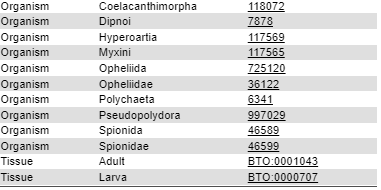


**EXTRACT**:

EXTRACT is a browser extension that identifies genes/proteins, chemical compounds, organisms, environments, tissues, diseases, phenotypes and Gene Ontology terms mentioned in a given piece of text and maps them to their corresponding ontology/taxonomy entries.

An example of extract usage:





**SpaCy:**

Spacy is an open-source software python library used in advanced natural language processing and machine learning. It will be used to build information extraction, natural language understanding systems, and to pre-process text for deep learning. It provides a lot of in-built functionalities, including deep neural networks.

For the installation (you can see <https://spacy.io/usage>), Python and pip are required.  
Commands for installation on windows based on the accuracy:

pip install -U pip setuptools wheel

pip install -U spacy

python -m spacy download en\_core\_web\_trf

python -m spacy download el\_core\_news\_lg

(for dispacy visualization (e.g displacy.serve(doc, style="ent")), the server provided is localhost:5000)

in order to make the dict for life\_stages.csv I used <https://products.groupdocs.app/conversion/html-to-csv> for html page (<https://www.marinespecies.org/traits/wiki/Traits:Lifestage>[)](https://www.marinespecies.org/traits/wiki/Traits:Lifestage),then) , then copied the stages and in libreoffice using function concat and hyperlink I made the links.(see cells)  
  
for body\_size.csv in order to extract links from html I used also this : http://tools.buzzstream.com/link-building-extract-urls

**Brat:**

Brat is a web-based tool for annotation visualization and editing. The tool is freely available and open source. Brat is designed in particular for structured annotation, where the notes are not free form text but have a fixed form that can be automatically processed and interpreted by a computer. The brat server is implemented in Python, and requires version 2.5.

(The online environment is not working. )

Installed in a standalone server: (needs Linux environment, wsl used)

commands:  
./install.sh -u  
python2 standalone.py (requires python2)  
  
To add data you have to put the .txt files in the data folder and then create an empty .ann file for each .txt.  
For the configuration files see [brat config](https://brat.nlplab.org/configuration.html#configuration-basics)  
Each annotation project typically defines its own annotation.conf (where you place: entities, relations, events, attributes). Defining visual.conf, tools.conf and kb\_shortcuts.conf is not necessary, and the system falls back on simple default visuals, tools and shortcuts if these files are not present.

Note:kapws mporeis na valeis tools gia automatic annotation.

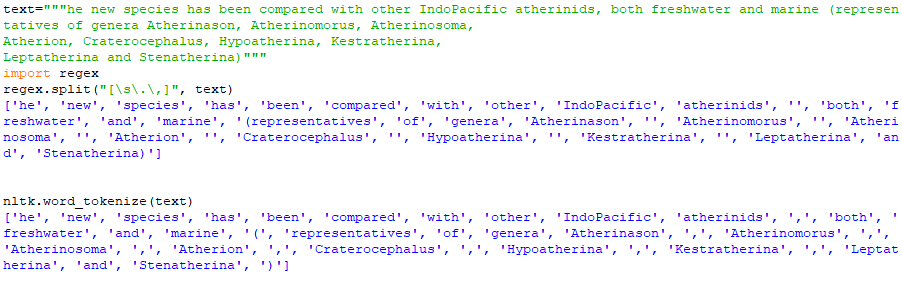
**Tagger:**

can’t compile, possible error in code package

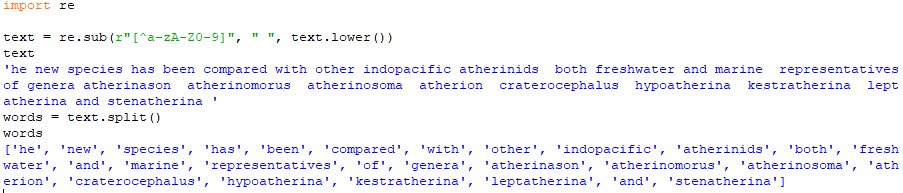
**NLTK**:

installation:  
python -m pip install nltk == 3.5

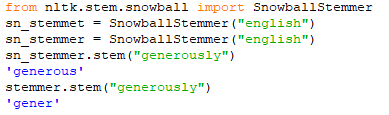
to download collections/models/corpora  
import nltk   
nltk.download()   
  
**tokenization** example, as it is shown, e.g the ‘(‘ or the spaces, the nltk.word\_tokenize is better



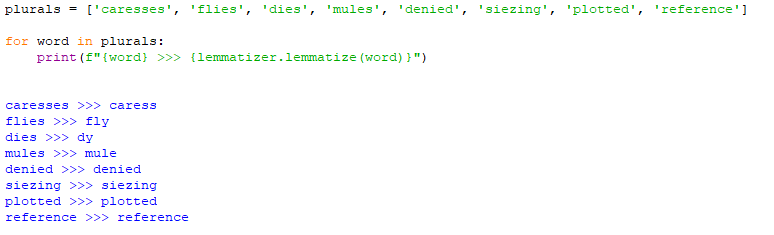
For **lower case** conversion:



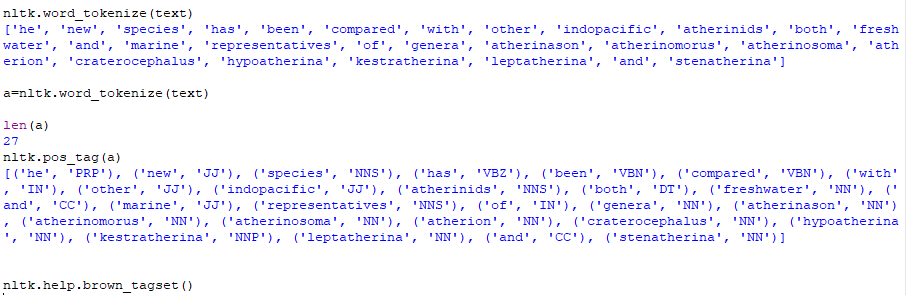
**Stemming**:  
snowballStemmer and porterStemmer are similar but snowball most of the time seems to have better results.



**Lemmatization**:



**Pos tags:**



nltk.help.brown\_tagset() gives the list of tags explained.

# TextBlob — great library for getting started

[TextBlob](https://textblob.readthedocs.io/en/dev/) is based on NLTK and Pattern. It has great API for all the common NLP operations. It’s a more practical library concentrated on day-to-day usage.

It’s great for initial prototyping in almost every NLP project. Unfortunately, it inherits the low performance from NLTK and therefore it’s not good for large scale production usage.

# TextBlob functionalities

tokenization, POS, NER, classification, sentiment analysis, spellcheck, parsing

# Pros

* easy to use and intuitive interface to NLTK
* provides language translation and detection which is powered by Google Translate

# Cons

* slow
* no neural network models
* no integrated word vectors

https://www.softkraft.co/python-nlp-libraries-features-us-cases-pros-and-cons/