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
```

last step: adding C:\bin directory to PATH environment variable.

When you run gnfinder command for the first time, it will create a 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
```

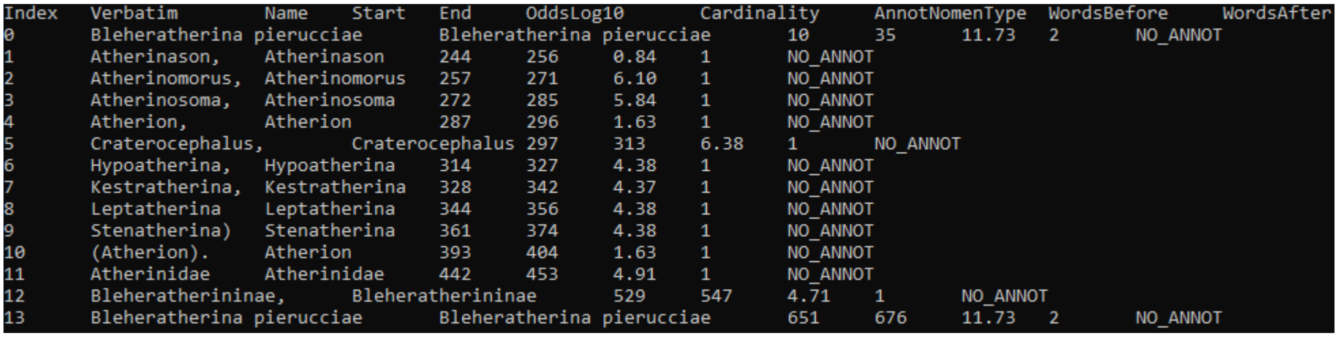
[**A test**

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:



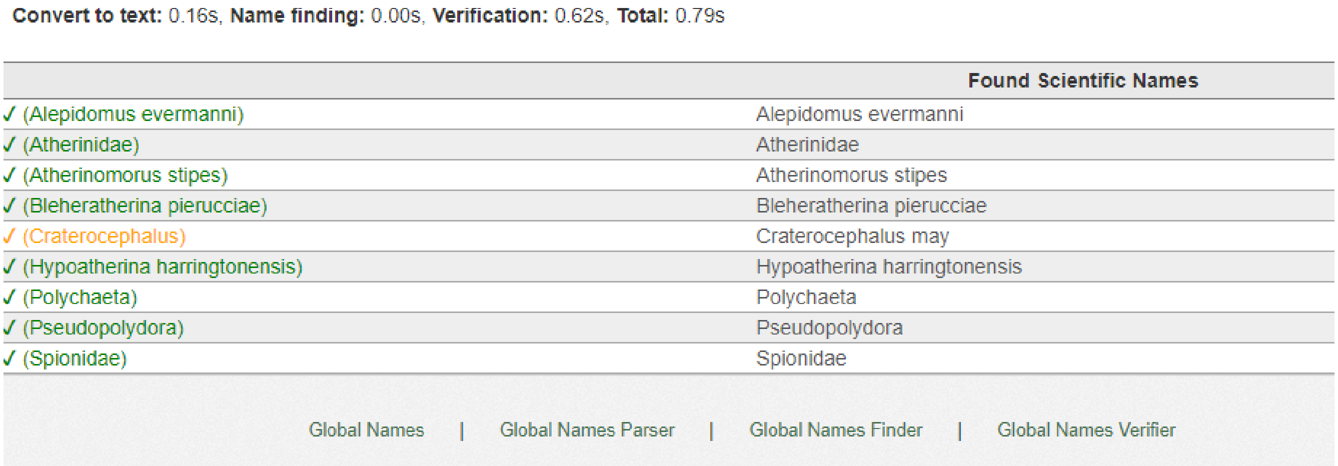
```
Index   Verbatim        Name    Start   End     OddsLog10       Cardinality     AnnotNomenType  WordsBefore     WordsAfter
0       Bleheratherina pierucciae       Bleheratherina pierucciae       10      35      11.73   2       NO_ANNOT
1       Atherinason,    Atherinason     244     256     0.84    1       NO_ANNOT
2       Atherinomorus,  Atherinomorus   257     271     6.10    1       NO_ANNOT
3       Atherinosoma,   Atherinosoma    272     285     5.84    1       NO_ANNOT
4       Atherion,       Atherion        287     296     1.63    1       NO_ANNOT
5       Craterocephalus,        Craterocephalus 297     313     6.38    1       NO_ANNOT
6       Hypoatherina,   Hypoatherina    314     327     4.38    1       NO_ANNOT
7       Kestratherina,  Kestratherina   328     342     4.37    1       NO_ANNOT
8       Leptatherina    Leptatherina    344     356     4.38    1       NO_ANNOT
9       Stenatherina)   Stenatherina    361     374     4.38    1       NO_ANNOT
10      (Atherion).     Atherion        393     404     1.63    1       NO_ANNOT
11      Atherinidae     Atherinidae     442     453     4.91    1       NO_ANNOT
12      Bleheratherininae,      Bleheratherininae       529     547     4.71    1       NO_ANNOT
13      Bleheratherina pierucciae       Bleheratherina pierucciae       651     676     11.73   2       NO_ANNOT
```

(the online public gnfinder didn't catch "(Atherion).")

]

It was also tested for url and pdf.

Verification example:



Convert to text: 0.16s, **Name finding:** 0.00s, **Verification:** 0.62s, **Total:** 0.79s

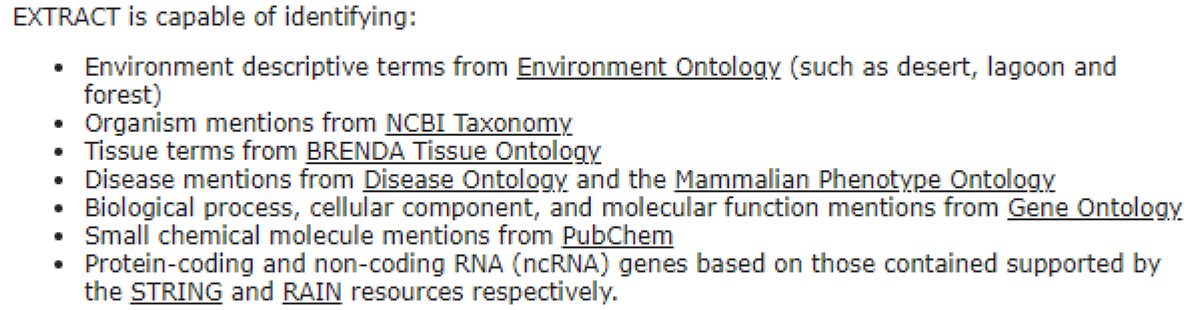
	Found Scientific Names
✓ (Alepidomus evermanni)	Alepidomus evermanni
✓ (Atherinidae)	Atherinidae
✓ (Atherinomorus stipes)	Atherinomorus stipes
✓ (Bleheratherina pierucciae)	Bleheratherina pierucciae
✓ (Craterocephalus)	Craterocephalus may
✓ (Hypoatherina harringtonensis)	Hypoatherina harringtonensis
✓ (Polychaeta)	Polychaeta
✓ (Pseudopolydora)	Pseudopolydora
✓ (Spionidae)	Spionidae

Global Names | Global Names Parser | Global Names Finder | Global Names Verifier

EXTRACT:

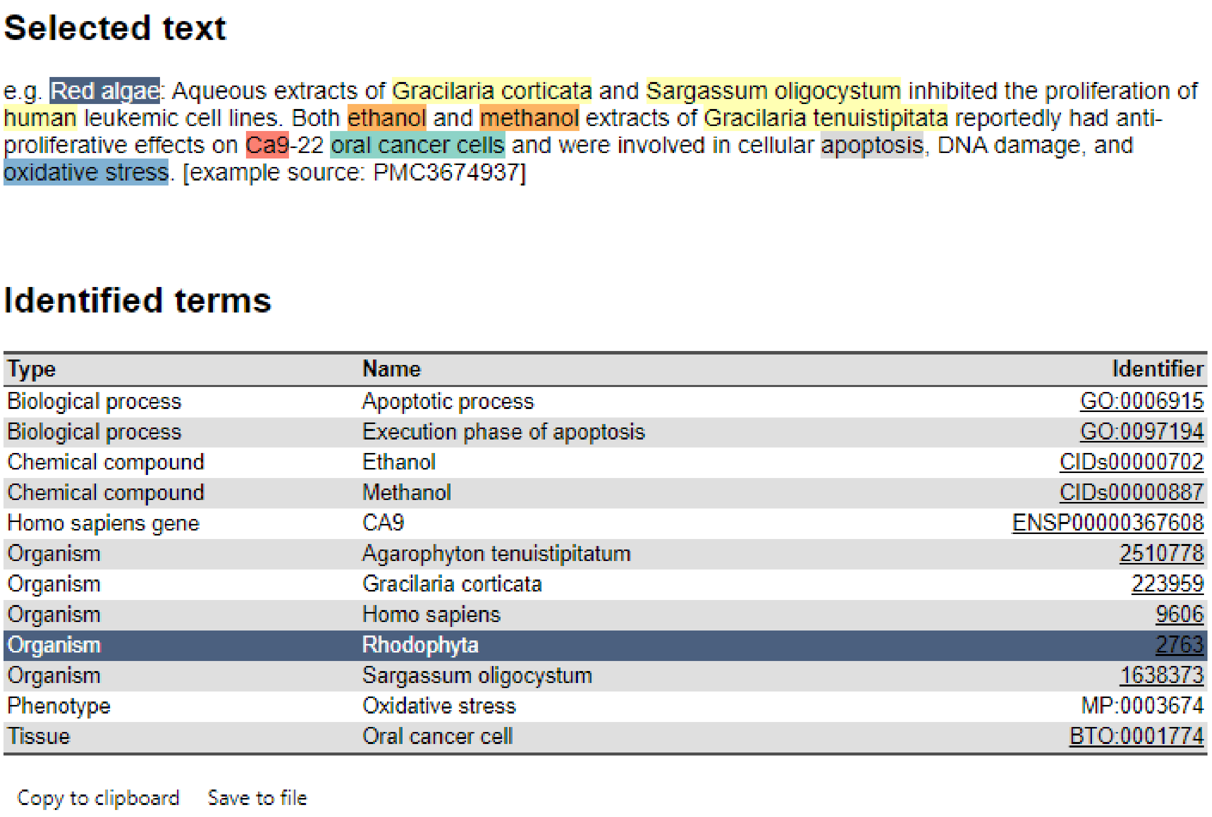
EXTRACT is a browser extension that identifies genes/proteins, chemical compounds, organisms, environments, tissues, diseases, phenotypes and Gene Ontology terms mentioned in a given piece of text and maps them to their corresponding ontology/taxonomy entries.

An example of extract usage:



EXTRACT is capable of identifying:

- Environment descriptive terms from Environment Ontology (such as desert, lagoon and forest)
- Organism mentions from NCBI Taxonomy
- Tissue terms from BRENDA Tissue Ontology
- Disease mentions from Disease Ontology and the Mammalian Phenotype Ontology
- Biological process, cellular component, and molecular function mentions from Gene Ontology
- Small chemical molecule mentions from PubChem
- Protein-coding and non-coding RNA (ncRNA) genes based on those contained supported by the STRING and RAIN resources respectively.

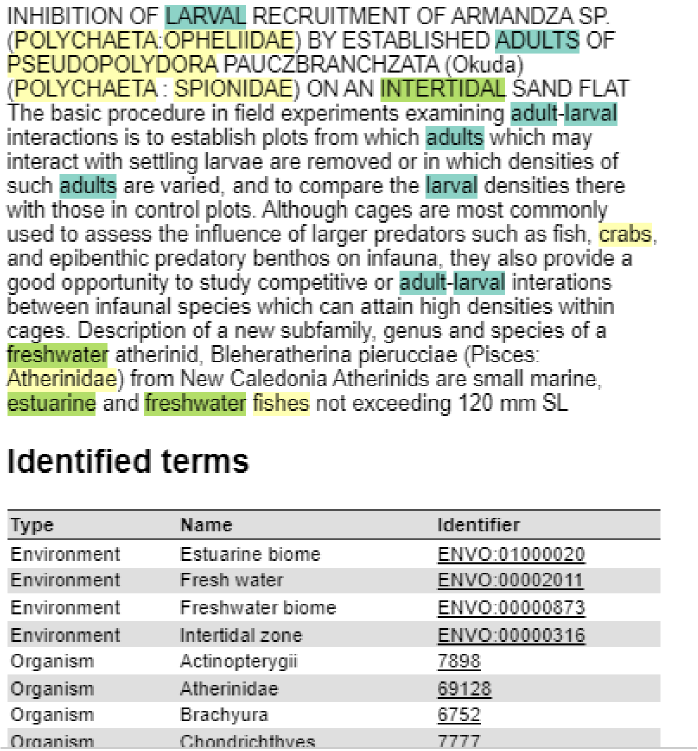


Selected text

e.g. Red algae: Aqueous extracts of Gracilaria corticata and Sargassum oligocystum inhibited the proliferation of human leukemic cell lines. Both ethanol and methanol extracts of Gracilaria tenuistipitata reportedly had anti-proliferative effects on Ca9-22 oral cancer cells and were involved in cellular apoptosis, DNA damage, and oxidative stress. [example source: PMC3674937]

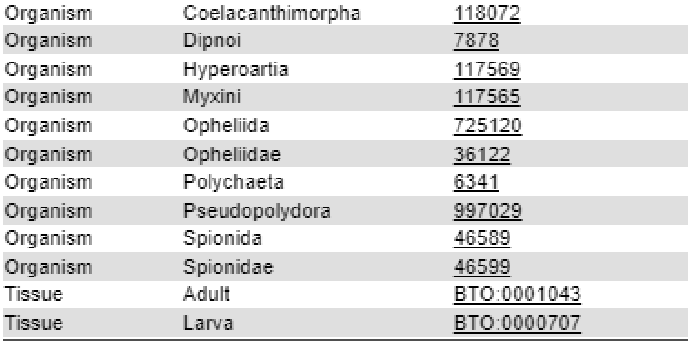
Identified terms

Type	Name	Identifier
Biological process	Apoptotic process	GO:0006915
Biological process	Execution phase of apoptosis	GO:0097194
Chemical compound	Ethanol	CIDs00000702
Chemical compound	Methanol	CIDs00000887
Homo sapiens gene	CA9	ENSP00000367608
Organism	Agarophyton tenuistipitatum	2510778
Organism	Gracilaria corticata	223959
Organism	Homo sapiens	9606
Organism	Rhodophyta	2763
Organism	Sargassum oligocystum	1638373
Phenotype	Oxidative stress	MP:0003674
Tissue	Oral cancer cell	BTO:0001774



INHIBITION OF LARVAL RECRUITMENT OF ARMANDZA SP. (POLYCHAETA:OPHELIIDAE) BY ESTABLISHED ADULTS OF PSEUDOPOLYDORA PAUCZBRANCHZATA (Okuda) (POLYCHAETA : SPIONIDAE) ON AN INTERTIDAL SAND FLAT The basic procedure in field experiments examining adult-larval interactions is to establish plots from which adults which may interact with settling larvae are removed or in which densities of such adults are varied, and to compare the larval densities there with those in control plots. Although cages are most commonly used to assess the influence of larger predators such as fish, crabs, and epibenthic predatory benthos on infauna, they also provide a good opportunity to study competitive or adult-larval interations between infaunal species which can attain high densities within cages. Description of a new subfamily, genus and species of a freshwater atherinid, Bleheratherina pierucciae (Pisces: Atherinidae) from New Caledonia Atherinids are small marine, estuarine and freshwater fishes not exceeding 120 mm SL

Identified terms



Type	Name	Identifier
Environment	Estuarine biome	ENVO:01000020
Environment	Fresh water	ENVO:00002011
Environment	Freshwater biome	ENVO:00000873
Environment	Intertidal zone	ENVO:00000316
Organism	Actinopterygii	7898
Organism	Atherinidae	69128
Organism	Brachyura	6752
Organism	Chondrichthyes	7777
Organism	Coelacanthimorpha	118072
Organism	Dipnoi	7878
Organism	Hyperoartia	117569
Organism	Myxini	117565
Organism	Opheliida	725120
Organism	Opheliidae	36122
Organism	Polychaeta	6341
Organism	Pseudopolydora	997029
Organism	Spionida	46589
Organism	Spionidae	46599
Tissue	Adult	BTO:0001043
Tissue	Larva	BTO:0000707

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) , 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
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:

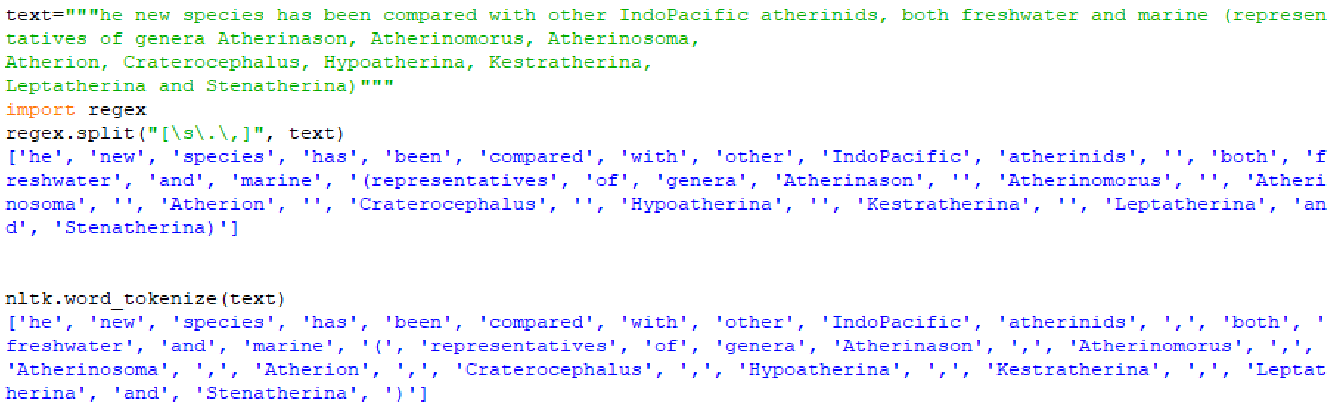
can’t compile, possible error in code package

NLTK:

installation:
python -m pip install nltk == 3.5

to download collections/models/corpora
import nltk
nltk.download()

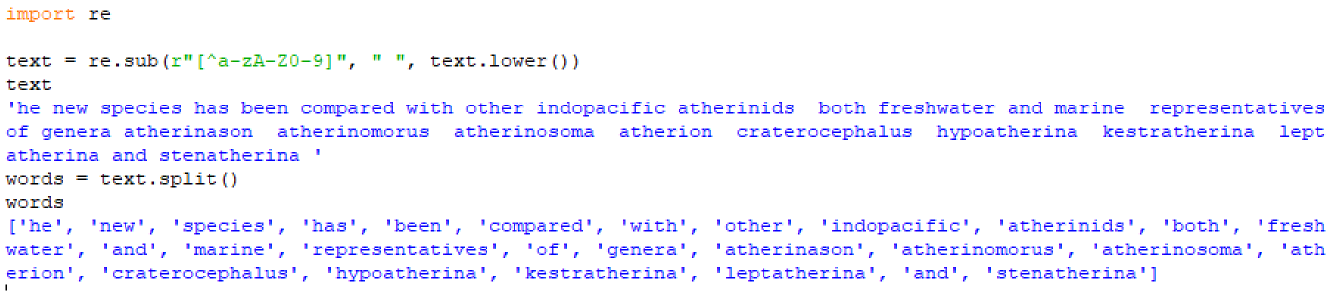
tokenization example, as it is shown, e.g the ‘(‘ or the spaces, the nltk.word_tokenize is better



```
text="""he new species has been compared with other IndoPacific atherinids, both freshwater and marine (represen
tatives of genera Atherinason, Atherinomorus, Atherinosoma,
Atherion, Craterocephalus, Hypoatherina, Kestratherina,
Leptatherina and Stenatherina)"""
import regex
regex.split("[\s\.\,]", text)
['he', 'new', 'species', 'has', 'been', 'compared', 'with', 'other', 'IndoPacific', 'atherinids', '', 'both', 'f
reshwater', 'and', 'marine', '(representatives', 'of', 'genera', 'Atherinason', '', 'Atherinomorus', '', 'Atheri
nosoma', '', 'Atherion', '', 'Craterocephalus', '', 'Hypoatherina', '', 'Kestratherina', '', 'Leptatherina', 'an
d', 'Stenatherina)']
```

```
nltk.word_tokenize(text)
['he', 'new', 'species', 'has', 'been', 'compared', 'with', 'other', 'IndoPacific', 'atherinids', ',', 'both', '
freshwater', 'and', 'marine', '(', 'representatives', 'of', 'genera', 'Atherinason', ',', 'Atherinomorus', ',',
'Atherinosoma', ',', 'Atherion', ',', 'Craterocephalus', ',', 'Hypoatherina', ',', 'Kestratherina', ',', 'Leptat
herina', 'and', 'Stenatherina', ')']
```

For **lower case** conversion:

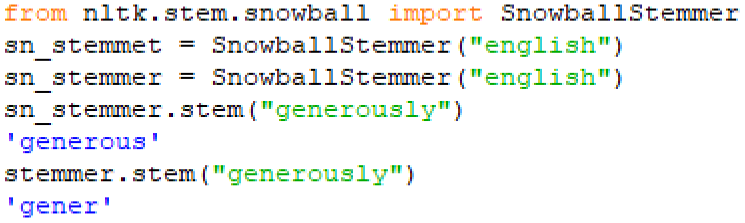


```
import re

text = re.sub(r"[^a-zA-Z0-9]", " ", text.lower())
text
'he new species has been compared with other indopacific atherinids  both freshwater and marine  representatives
of genera atherinason  atherinomorus  atherinosoma  atherion  craterocephalus  hypoatherina  kestratherina  lept
atherina and stenatherina '
words = text.split()
words
['he', 'new', 'species', 'has', 'been', 'compared', 'with', 'other', 'indopacific', 'atherinids', 'both', 'fresh
water', 'and', 'marine', 'representatives', 'of', 'genera', 'atherinason', 'atherinomorus', 'atherinosoma', 'ath
erion', 'craterocephalus', 'hypoatherina', 'kestratherina', 'leptatherina', 'and', 'stenatherina']
```

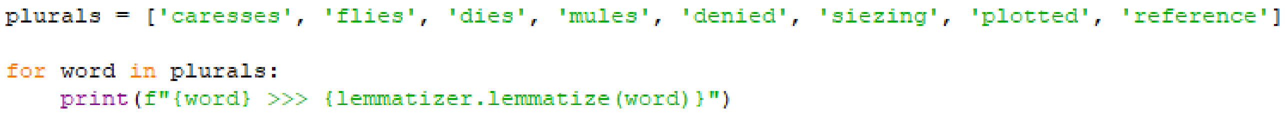
Stemming:

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



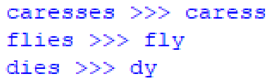
```
from nltk.stem.snowball import SnowballStemmer
sn_stemmet = SnowballStemmer("english")
sn_stemmer = SnowballStemmer("english")
sn_stemmer.stem("generously")
'generous'
stemmer.stem("generously")
'gener'
```

Lemmatization:



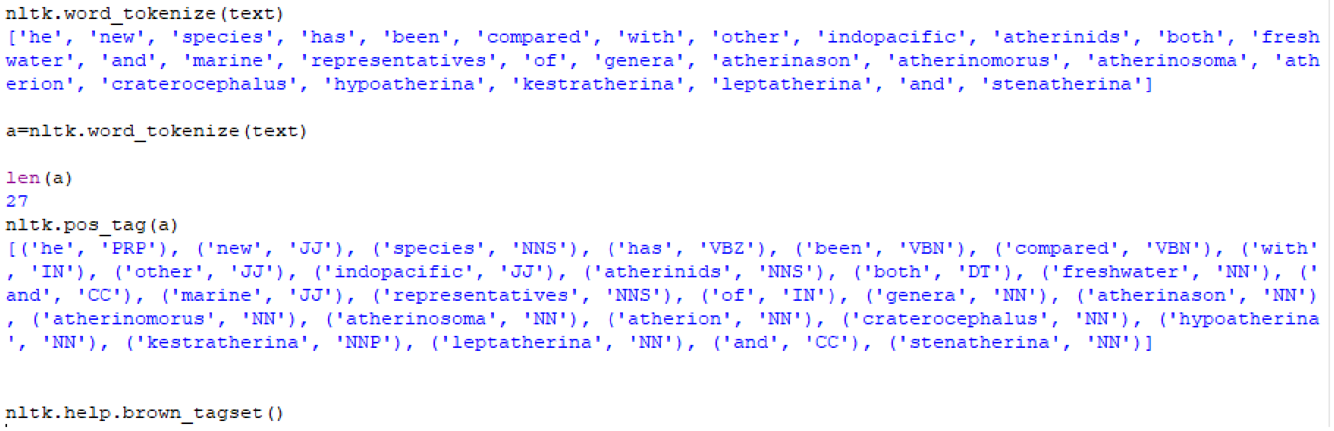
```
plurals = ['caresses', 'flies', 'dies', 'mules', 'denied', 'siezing', 'plotted', 'reference']

for word in plurals:
    print(f"{word} >>> {lemmatizer.lemmatize(word)}")
```



```
caresses >>> caress
flies >>> fly
dies >>> dy
```

Pos tags:



```
nltk.word_tokenize(text)
['he', 'new', 'species', 'has', 'been', 'compared', 'with', 'other', 'indopacific', 'atherinids', 'both', 'fresh
water', 'and', 'marine', 'representatives', 'of', 'genera', 'atherinason', 'atherinomorus', 'atherinosoma', 'ath
erion', 'craterocephalus', 'hypoatherina', 'kestratherina', 'leptatherina', 'and', 'stenatherina']

a=nltk.word_tokenize(text)

len(a)
27
nltk.pos_tag(a)
[('he', 'PRP'), ('new', 'JJ'), ('species', 'NNS'), ('has', 'VBZ'), ('been', 'VBN'), ('compared', 'VBN'), ('with'
, 'IN'), ('other', 'JJ'), ('indopacific', 'JJ'), ('atherinids', 'NNS'), ('both', 'DT'), ('freshwater', 'NN'), ('
and', 'CC'), ('marine', 'JJ'), ('representatives', 'NNS'), ('of', 'IN'), ('genera', 'NN'), ('atherinason', 'NN')
, ('atherinomorus', 'NN'), ('atherinosoma', 'NN'), ('atherion', 'NN'), ('craterocephalus', 'NN'), ('hypoatherina
', 'NN'), ('kestratherina', 'NNP'), ('leptatherina', 'NN'), ('and', 'CC'), ('stenatherina', 'NN')]


nltk.help.brown_tagset()
```

nltk.help.brown_tagset() gives the list of tags explained.

TextBlob — great library for getting started

TextBlob 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/