Theseus Documentation

Release 0.7

David Rodrigues

October 25, 2010

CONTENTS

Latest release: 0.7

Latest update: October 25, 2010

CHAPTER

ONE

CONTENTS

1.1 What is Theseus?

A brief overview on what Theseus is, and is not

Theseus is a python package that includes several modules to deal with webpage retrival and text processing.

It is the baisis for the deployment of a system based on bash scripts and python

It his the responsible for collecting and processing newspaper pages for the observatorium (see details at http://theobservatorium.eu)

1.2 What will Theseus be at version 1.0?

Theseus will end up (eventually) being made of 4 groups of programs/modules/scripts inside the theseus package

- **Crawler** (for online gathering of news items)
- **Processor** (for processing of textual data)
- **Utils** (acessory methods and utilities for pre and post processing)
- **Examples** (To help users to start using **theseus**)

1.3 Theseus now!

Theseus is a package of several python modules including:

- processor
- crawler
- utils
- examples

Theseus is part of http://theobservatorium.eu project.

1.3.1 theseus.processor

theseus.py

A Python Library for text processing in The Observatorium project

Visit http://theobservatorium.eu/ for the latest version

References

class theseus.processor.theseus.**Channel**(*label*)
A Channel contains all documents of a certain channel

- *label* is a string
- *doc* is a DocNode

class theseus.processor.theseus.**DocNode**(*idn=''*, *fnm=''*, *txt=''*, *ttl=''*, *lang='en'*)
The DocNode is the basic structue that olds each document in a corpus

- *idn* Id number of the node
- *fnm* File name of the Document
- *txt* Text of the Document
- *ttl* Time to Live
- *lang='en'* The language of the text, defaults to english

extractSentences()
Extract all the sentences of the document

class theseus.processor.theseus.**Domain**(*label*, *words*=[])
A Domain is a field with a collection of words and a label

Domain words should all be lower capital and without stopwords!

class theseus.processor.theseus.**Sentence**(*text*, *lang='en'*)
The Sentence is one of the building blocks of Documents

cleanText()

Processes the raw text of a sentence:

- creates a `cleaned` text without unauthorized letters,
- creates a `words` list
- creates a `cleanedWords` list without **stopwords**

theseus.processor.theseus.**binary**(*token*, *doc*)
Calculates the Binary existence of a token in a document (doc)

returns 1 if token exists, 0 otherwise

- *token* is a string ex. 'word'
- *doc* is a `list` ex. ['this' 'is' 'a' 'word' 'document']

theseus.processor.theseus.**cleanString**(*s1*)
Cleans strings from unauthorized letters

theseus.processor.theseus.**cleanStringNoDel**(*s1*)
Cleans strings from unauthorized letters

theseus.processor.theseus.**clusterHist**(*clst*)
Takes a List of DocNodes and returns an histogram of the most common words

theseus.processor.theseus.**dtf**(*token*, *corpus*)
Calculates the fraction of documents of the corpus that have a token

- *token* is a string ex. 'word'
- *corpus* is a `list` of `lists` ex. [['this' 'is' 'a' 'word' 'document']['document' 'two']]

theseus.processor.theseus.**enClean**()
English Stop Words

theseus.processor.theseus.**esClean**()
Spanish Stop Words

obtained from http://www.ranks.nl/stopwords/spanish.html

theseus.processor.theseus.**extractPhrases**(*s1*)
extractPhrases breaks a document into a sequence of phrases.

XXX: We need to deal with numbers...

theseus.processor.theseus.**frClean**()
Frenc Stop Words

obtained from http://www.ranks.nl/stopwords/french.html

theseus.processor.theseus.**idf**(*token*, *corpus*)
Calculates the inverse document frequency of a token

- *token* is a string ex. 'word'
- *corpus* is a `list` of `lists` ex. [['this' 'is' 'a' 'word' 'document']['document' 'two']]

theseus.processor.theseus.**jaccard**(*s1*, *s2*)
Calculates de jaccard index for two lists

theseus.processor.theseus.**logtf**(*token*, *doc*)
Calculates the Log Term Frequency in a certain document (doc)

- *token* is a string ex. 'word'
- *doc* is a `list` ex. ['this' 'is' 'a' 'word' 'document']

theseus.processor.theseus.**logtfidf**(*token*, *doc*, *corpus*)
Calculates the Log Term Frequency-Inverse Document Frequency of a token

- *token* is a string ex. 'word'
- *doc* is a `list` ex. ['this' 'is' 'a' 'word' 'document']
- *corpus* is a `list` of `lists` ex. [['this' 'is' 'a' 'word' 'document']['document' 'two']]

theseus.processor.theseus.**normF**(*token*, *channel*)
Calculates the normalized frequency of a term in a channel of documents

see `tfpdf()`

theseus.processor.theseus.**ptClean**()
Portuguese Stop Words

theseus.processor.theseus.**tf**(*token*, *doc*)
Calculates the term frequency in a certain document (doc)

- *token* is a string ex. 'word'

•*doc* is a `list` ex. ['this' 'is' 'a' 'word' 'document']

`theseus.processor.theseus.`**`tfidf`**(*token*, *doc*, *corpus*)
Calculates the Term Frequency-Inverse Document Frequency of a token

•*token* is a string ex. 'word'

•*doc* is a `list` ex. ['this' 'is' 'a' 'word' 'document']

•*corpus* is a `list` of `lists` ex. [['this' 'is' 'a' 'word' 'document']['document' 'two']]

`theseus.processor.theseus.`**`tfpdf`**(*token*, *channels*)
Calculates the Term Frequency * Proportional Document Frequency (TF*PDF) [Bun2006] [Ishzuka2001] [Ishzuka2002]

•*token* is a string

•*channels* is a `list` of `Channel`

text2tag.py

Converts HTML files to TXT according to the Text to Tag ratio proposed by [Weninger2008]

Usage: `$ python text2tag.py <inputfile> <smooth-radius>`

References

class `theseus.processor.text2tag.`**`MyStripper`**
docstring for MyStripper

`handle_data`(*d*)
docstring for handle_data

`theseus.processor.text2tag.`**`process`**(*infile*)
Process the html infile

`theseus.processor.text2tag.`**`rm_blank_lines`**(*string*)
removes blank lines from a string

`theseus.processor.text2tag.`**`rm_head`**(*string*)
rm_head: Removes everything in the <head></head> of the document

`theseus.processor.text2tag.`**`rm_scripts`**(*string*)
rm_scripts: Removes <scripts> and Comments <!– –> from html files

`theseus.processor.text2tag.`**`rm_tags`**(*string*)
rm_tags: Removes HTML Tags from string

1.3.2 theseus.crawler

getUrl.py

Parses a downloaded RSS file and downloads all items in the Feed

`theseus.crawler.getUrl.`**`main`**()
This module receives as argument the folder where the rss.xml file is stored

1.3.3 theseus.utils

dnet.py

a Telnet Library to Connect to Guess

Guess is a graph exploration tool that can be found at: http://guess.wikispot.org/Front_Page

class `theseus.utils.dnet.`**`Dnet`**(*gr*)
Constructor for a telnet library to connect to Guess See Guess - http://graphexploration.cond.org/

`send`(*cmd*)
Send a message to Guess via telnet. Guess - http://graphexploration.cond.org/

cleanDuplicates.py

Clean files with the same contents from a folder, keeping the oldest one.

usage: `$ python cleanDuplicates.py DIR_TO_CLEAN`

`theseus.utils.cleanDuplicates.`**`main`**()
receives as argument the dir to clean

1.3.4 theseus.examples

See *Roadmap* for details

1.4 Download Theseus

1.4.1 Version 0.7

- theseus-0.7 http://theobservatorium.eu/zips/theseus-0.7.zip

1.5 How to

Some simple examples to get you started using python and Theseus

1.5.1 Process 11 TXT files inside a "TXT" folder

eccs10bursaries.py

This example will demonstrate the use of Theseus to process a set of texts that are archived in a folder ./TXT

Text Files are named 01.txt ... 11.txt

`theseus.examples.eccs10bursaries.`**`main`**()
This example will demonstrate the use of Theseus to process a set of texts that are archived in a folder ./TXT

Text Files are named 01.txt ... 11.txt

Check the source code to detailed step by step instructions

1.6 Frequently Asked Questions (FAQ)

Common questions and answers for common (sometimes) problems

1.7 Contact The Observatorium

The Observatorium Webstie is at http://www.theobservatorium.eu

David Rodrigues email is m4467@iscte.pt

1.8 Roadmap

1.8.1 0.8

- add `utils.py` and collect some dispersed scripts into this package. **[Done]**
- **solve abrveviation problems in the identification of phrases** ex. "His name is D. Rodrigues and he his a scientist". The dot after D will break a sentence. So one needs to be awere of this. Another problem is that of the use of hiffens. a "pre-conference" should be treated as 1 word and not as two. This things have to be processed at the Document level before breaking the Document into Sentences

1.8.2 0.7 Present Version

- add `fr` stop words **[Done]**
- add `es` stop words **[Done]**
- rename **theseus** module to **processor** and incorporate `crawler.py` code into *Theseus* as **crawler** module **[Done]**
- **Documentation** Write what is Thesues section of this documentation. **[Done]**

1.8.3 0.6

- implement `theseus.tfpdf()` method **[Done]**
- test `theseus.tfpdf()` with text from ECCS'10 Bursaries **[Done]**
- **Documentation** Write the ECCS'10 Bursaries text as an example of usage. **[Done]**

1.8.4 0.5.1

CHAPTER

TWO

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

BIBLIOGRAPHY

[Bun2006] Bun, K., & Ishizuka, M. (2006). Emerging topic tracking system in WWW. Knowledge-Based Systems, 19(3), 164-171. doi: 10.1016/j.knosys.2005.11.008.

[Ishzuka2001] Ishizuka, M. (n.d.). Topic extraction from news archive using TF*PDF algorithm. Proceedings of the Third International Conference on Web Information Systems Engineering, 2002. WISE 2002., 73-82. IEEE Comput. Sci. doi: 10.1109/WISE.2002.1181645.

[Ishzuka2002] Ishizuka, M. (2001). Emerging Topic Tracking System. Proceedings Third International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems. WECWIS 2001, 2-11. IEEE Comput. Soc. doi: 10.1109/WECWIS.2001.933900

[Weninger2008] Weninger, T., & Hsu, W. H. (2008). Text Extraction from the Web via Text-to-Tag Ratio. 2008 19th International Conference on Database and Expert Systems Applications, 23-28. Ieee. doi: 10.1109/DEXA.2008.12.

PYTHON MODULE INDEX

t

INDEX