Theseus Documentation

Release 0.6.0rc

David Rodrigues

October 21, 2010

CONTENTS

Latest release: 0.6.0rc

Latest update: October 21, 2010

CHAPTER

ONE

CONTENTS

1.1 What is Theseus?

A brief overview on what Theseus is, and is not

It's a collection of scripts, programs and libraries

mainly consist of bash scripts + python scripts

It his the responsible for collecting and processing newspaper pages for the observatorium

1.2 What will Theseus be at version 1.0?

Theseus will end up (eventually) being made of 3 groups of programs/scripts:

- **Crawler** (for online gathering of news items)
- **Processor** (for processing of textual data)
- **Utils** (acessory methods and utilities for pre and post processing)

1.3 Theseus now!

The Theseus Project is a collection of several scripts that help the scientist to manipulate text documents in manner to extract useful information.

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

This is the main module for data processing. It's where several classes that old the data are defined.

1.3.1 theseus.py

A Python Library for text processing in The Observatorium project

http://theobservatorium.eu/

Created by David Rodrigues on 2010-02-03.

Copyright (c) 2010 Sixhat Pirate Parts. All rights reserved.

class `theseus.`**`Channel`**(*label*)

A Channel contains all documents of a certain channel

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

class `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.`**`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.`**`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.`**`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.`**`cleanString`**(*s1*)

Cleans strings from unauthorized letters

`theseus.`**`cleanStringNoDel`**(*s1*)

Cleans strings from unauthorized letters

`theseus.`**`clusterHist`**(*clst*)

Takes a List of DocNodes and returns an histogram of the most common words

`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.`**`enClean`**()

English Stop Words

`theseus.`**`extractPhrases`**(*s1*)

extractPhrases breaks a document into a sequence of phrases.

XXX: We need to deal with numbers...

`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.`**`jaccard`**(*s1*, *s2*)

Calculates de jaccard index for two lists

`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.`**`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.`**`normF`**(*token*, *channel*)

Calculates the normalized frequency of a term in a channel of documents

see `theseus.tfpdf()`

`theseus.`**`ptClean`**()

Portuguese Stop Words

`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.`**`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.`**`tfpdf`**(*token*, *channels*)

> Calculates the Term Frequency * Proportional Document Frequency (TF*PDF)
>
> - *token* is a string
> - *channels* is a `list` of `Channel`

References

1.3.2 crawler.py (will available in v.0.7)

1.3.3 utils.py (will available in v.0.8)

See *Roadmap* for details

1.4 How to

Some simple examples to get you started using python and Theseus

1.4.1 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

`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

1.5 Frequently Asked Questions (FAQ)

Common questions and answers for common (sometimes) problems

1.6 Contact The Observatorium

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

David Rodrigues email is m4467@iscte.pt

1.7 Roadmap

1.7.1 0.8

- add `utils.py` and collect some dispersed scripts into this package.
- **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

1.7.2 0.7

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

1.7.3 0.6 Present Version

- 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]**

1.7.4 0.5.1

CHAPTER

TWO

INDICES AND TABLES

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

PYTHON MODULE INDEX

INDEX