py-pdf-term

fully-configurable terminology extraction module written in Python.

https://github.com/kumachan-mis/py-pdf-term

Motivation

- There are countless number of implementations of terminology extraction
- Many of them support personal trials to demonstrate efficacy of the algorithm
- However, difficult to use in practical softwares

The goal is to support both of **laboratory use** and **practical use**

Features - for laboratory use

switch of ranking algorithms

- The more algorithms are proposed, the more essential selecting a suitable one is
- This module enables you to select a ranking algorithm with a configuration

plug-in for trial-and-error

- Trial-and-error is also essential to have ideal outputs
- This module enables you to plug your ideas into the process as classes

Features - for practical use

i/o customizing

- In practical softwares, files are often not in a local storage.
- I/O functions for a remote storage are more complicated than built-in ones
- This module enables you to replace I/O functions with yours

cache mechanism

- In practical softwares, performance is one of the biggest problems
- This module has cache mechanism to save intermediate output

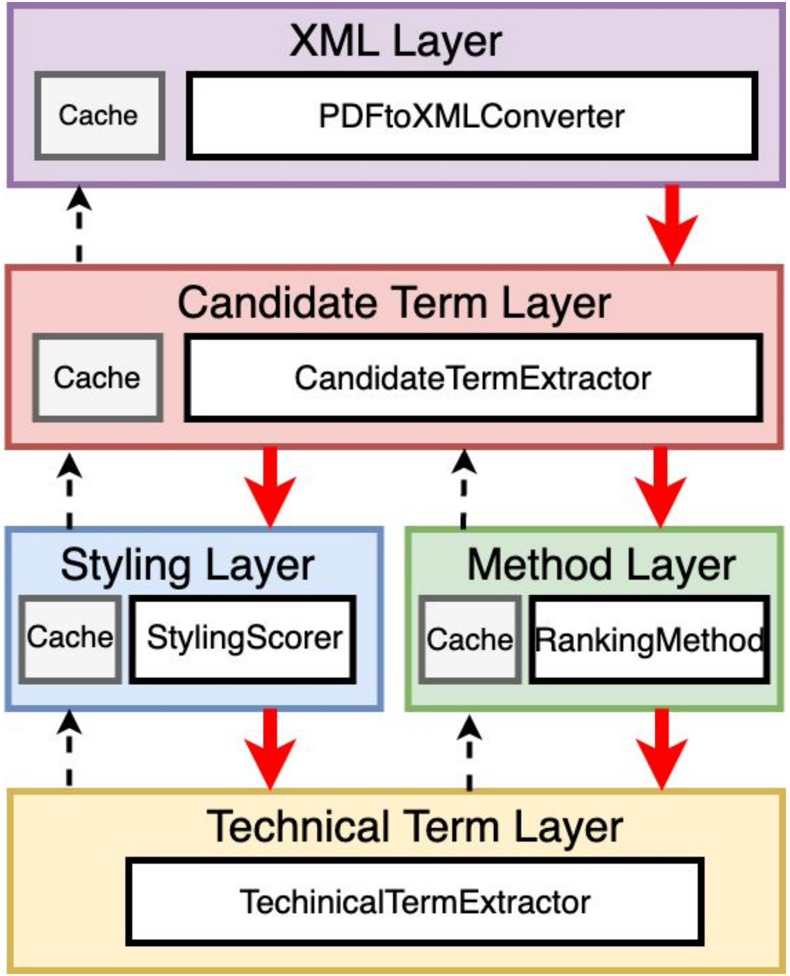
5-layers architecture

XML Layer

- Converts a PDF file to a XML file
- Depends on pdfminer.six

Candidate Term Layer

- Extracts candidates of terminologies
- Depends on spaCy



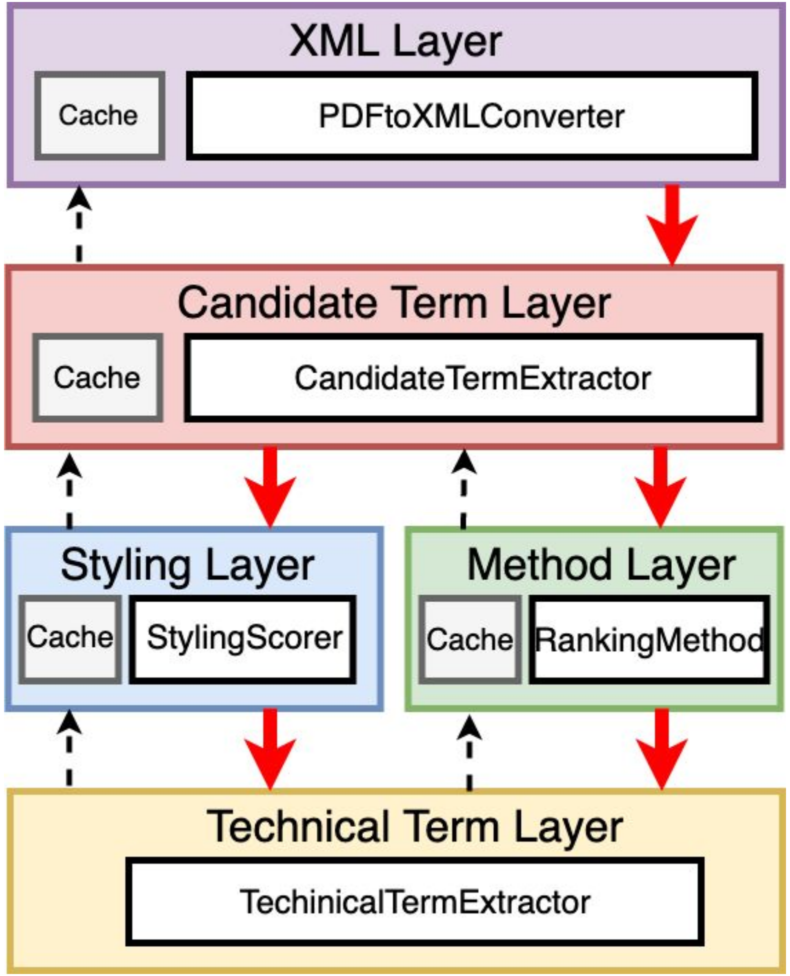
5-layers architecture

Method Layer

- Calculates method scores of candidates
- Frequency, colocation likelihood

Styling Layer

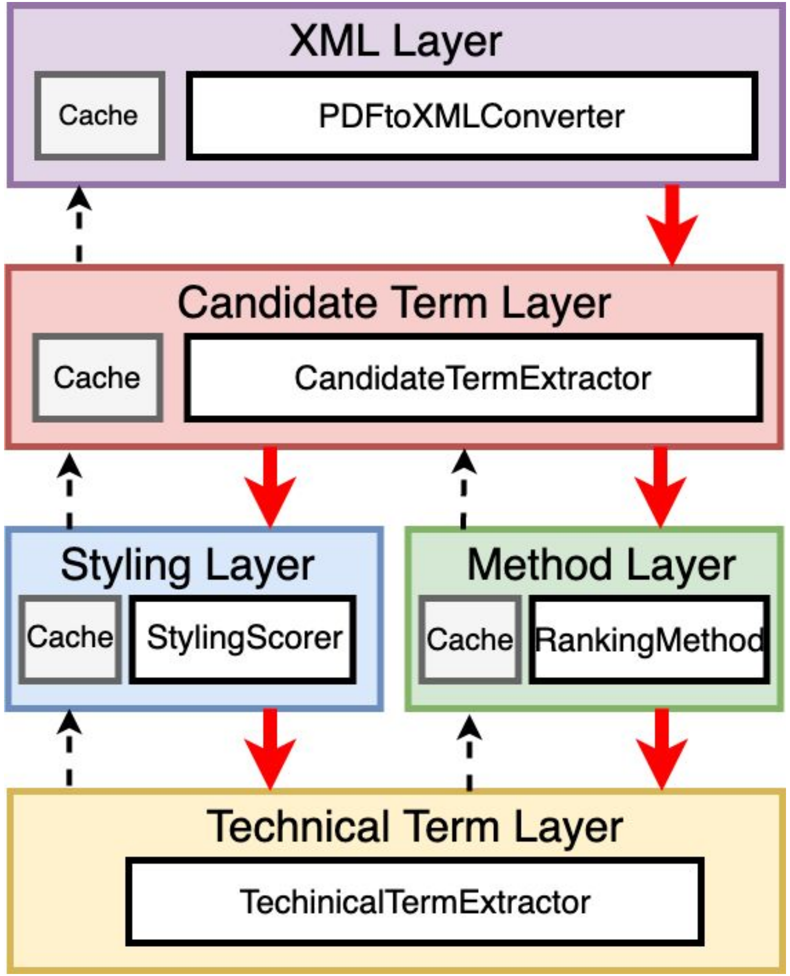
- Calculates styling scores of candidates
- Font size, font color



5-layers architecture

Technical Term Layer

- Selects terminologies from candidates
- The order of the terminologies is appearance order in a PDF file.


XML Layer
Cache
PDFtoXMLConverter
Candidate Term Layer
Cache
CandidateTermExtractor
Styling Layer
Cache
StylingScorer
Method Layer
Cache
RankingMethod
Technical Term Layer
TechinicalTermExtractor