

PaperParser

1. Search for and download papers in a relevant (clean energy) field
 - Test case: perovskite solar cells
2. Process and clean papers into a usable format
3. Search and parse papers for relevant information
 - Synthesis parameters
 - Performance metrics
4. Collect and clean relevant information into a usable format
5. Visualize relationships between synthetic parameters and device performance

PaperParser

1. Search for and download papers in a relevant (clean energy) field
 - Test case: perovskite solar cells
2. Process and clean papers into a usable format
3. Search and parse papers for relevant information
 - Synthesis parameters
 - Performance metrics
4. Collect and clean relevant information into a usable format
5. Visualize relationships between synthetic parameters and device performance

Packages

- spaCy
 - What: Organize text into useful categories
 - Why: Cleaning data (Step 4)
- ChemicalTagger
 - What: Find and organize chemistry-relevant parameters
 - Why: Search and parse (Step 3) (and 4?)
- ChemDataExtractor
 - What: Find chemistry-relevant parameters
 - Why: Search and parse (Step 3)

spaCy-NLP

- Free, open-source lib for adv NLP in python
- Fastest syntactic parser in the world
- Its accuracy is within 1% of the best available
- Supports 34+ languages
- Easy deep learning integration
- Easy model packaging and deployment
- Pre-trained statistical models and word vectors
- ~9000 commits and recent commits within a month
- Cited in research journals for Text extraction and machine learning in Materials Science domain

- <https://spacy.io/>
- Kim, E., Huang, K., Saunders, A., McCallum, A., Ceder, G., & Olivetti, E. (2017). *Materials Synthesis Insights from Scientific Literature via Text Extraction and Machine Learning*. *Chemistry of Materials*, 29(21), 9436–9444.

	SPACY	SYNTAXNET	NLTK	CORENLP
Programming language	Python	C++	Python	Java
Neural network models	✓	✓	✗	✓
Integrated word vectors	✓	✗	✗	✗
Multi-language support	✓	✓	✓	✓
Tokenization	✓	✓	✓	✓
Part-of-speech tagging	✓	✓	✓	✓
Sentence segmentation	✓	✓	✓	✓
Dependency parsing	✓	✓	✗	✓
Entity recognition	✓	✗	✓	✓
Coreference resolution	✗	✗	✗	✓

Input

v2.0.18 · Python 3 · via Binder

```
# pip install spacy
# python -m spacy download en_core_web_sm

import spacy

# Load English tokenizer, tagger, parser, NER and word vectors
nlp = spacy.load('en_core_web_sm')

# Process whole documents
text = (u"When Sebastian Thrun started working on self-driving cars at "
        u"Google in 2007, few people outside of the company took him "
        u"seriously. "I can tell you very senior CEOs of major American "
        u"car companies would shake my hand and turn away because I wasn't "
        u"worth talking to," said Thrun, now the co-founder and CEO of "
        u"online higher education startup Udacity, in an interview with "
        u"Recode earlier this week.")

doc = nlp(text)

# Find named entities, phrases and concepts
for entity in doc.ents:
    print(entity.text, entity.label_)

# Determine semantic similarities
doc1 = nlp(u"my fries were super gross")
doc2 = nlp(u"such disgusting fries")
similarity = doc1.similarity(doc2)
print(doc1.text, doc2.text, similarity)
```

Output

Sebastian Thrun PERSON

Google ORG

2007 DATE

American NORP

Thrun PERSON

Recode ORG

earlier this week DATE

my fries were super gross such disgusting fries 0.7139701576579747

Chemical Tagger

- How the package works?
 - Text normalization – preprocessing
 - Tokenization – split into elements
 - Tagging – chemical entities, chemistry related, parts-of-speech
 - Phrase parsing – assign structure to text
 - Action phrase identification – e.g. add, dissolve, stir
- Appeal
 - Separates out action steps, synthesis parameters/conditions etc. in a structured database output
- Drawbacks
 - Java

UNIVERSITY OF CAMBRIDGE

ChemicalTagger

University of Cambridge > Department of Chemistry > Centre for Molecular Informatics

To synthesize the precursors, we obtained precipitates by pouring toluene as a nonsolvent into 1.0 M PbI₂ solution dissolved in DMSO. The x-ray diffraction (XRD) pattern of the resulting complex [Fig. 1B(a)] matched that of the PbI₂(DMSO)₂ phase (5 , 20). The as-prepared PbI₂(DMSO)₂ was then annealed at 60 °C for 24 hours in vacuum to obtain PbI₂(DMSO) by removal of 1 mol DMSO. The XRD pattern of the vacuum-annealed powder [Fig. 1B(b)] did not match that of PbI₂(DMSO)₂, implying that the PbI₂(DMSO)₂ transformed into a different phase by releasing some DMSO molecules. The content of DMSO in the as-annealed powder was estimated by thermogravimetric analysis (TGA). TGA was suitable for this purpose because the only volatile species in the powder was DMSO. The TGA results of the PbI₂(DMSO)₂ and PbI₂(DMSO) complexes are shown in Fig. 1C. The PbI₂(DMSO)₂ complex exhibited a two-step decomposition process with weight loss of 12.6 % at each step, whereas the vacuum-annealed PbI₂(DMSO) complex showed a single-step decomposition. The decomposition of both the complexes was completed at the same temperature (138.6 °C). The powders obtained by vacuum-annealing PbI₂(DMSO)₂ complex at 60 °C can be regarded as one of the most thermodynamically stable forms among the various crystalline PbI₂(DMSO)-based complexes, which are similar to those of PbBr₂(DMSO) and PbCl₂(DMSO) (21). The DMSO content of the vacuum-annealed PbI₂(DMSO) complex was also checked by elemental analysis, which yielded H = 1.0 % (1.1 %), and C = 4.1 % (4.4 %), where the values expressed in parentheses indicate the theoretical mass percent for a given element in C₂H₆SOPbI₂.

Actions:
☒ Add ☒ Synthesize ☒ Dissolve ☒ Yield ☒ Precipitate ☒ Remove ☒ Wait

Conditions:
☐ TimePhrase ☐ TempPhrase

Molecules:
☐ Solvent ☐ Other

Phrases:
☐ NounPhrase ☐ PrepPhrase ☐ VerbPhrase

Quantitative_Terms:
☐ Quantity

[View XML](#)

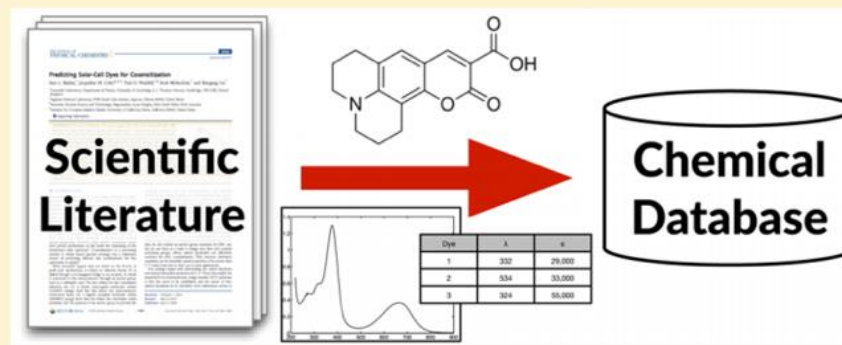
Links
→ Web Interface
→ Instructions
→ Documentation
→ Publication
→ BitBucket Project
→ Downloads

ChemDataExtractor: A Toolkit for Automated Extraction of Chemical Information from the Scientific Literature

Matthew C. Swain and Jacqueline M. Cole*

Cavendish Laboratory, University of Cambridge, J. J. Thomson Avenue, Cambridge, CB3 0HE, U.K.

“(A) toolkit for the **automated extraction of chemical entities and their associated properties, measurements, and relationships from scientific documents** that can be used to populate structured chemical databases.”



Characteristics of note;

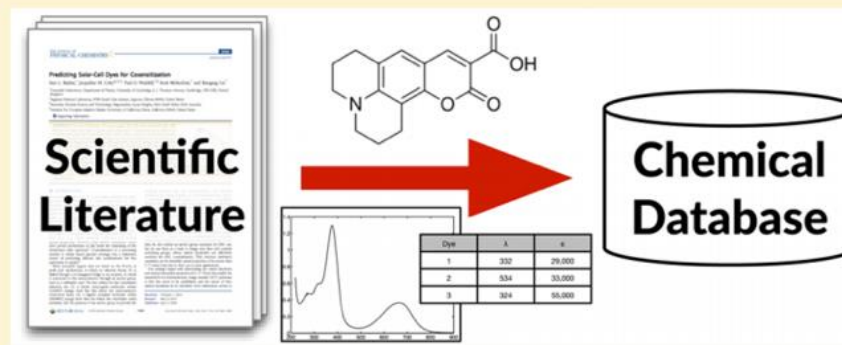
- chemistry-aware natural language processing pipeline
 - trained by unsupervised word clustering in a “massive corpus of chemistry articles”

ChemDataExtractor: A Toolkit for Automated Extraction of Chemical Information from the Scientific Literature

Matthew C. Swain and Jacqueline M. Cole*

Cavendish Laboratory, University of Cambridge, J. J. Thomson Avenue, Cambridge, CB3 0HE, U.K.

“(A) toolkit for the **automated extraction of chemical entities and their associated properties, measurements, and relationships from scientific documents** that can be used to populate structured chemical databases.”



Characteristics of note;

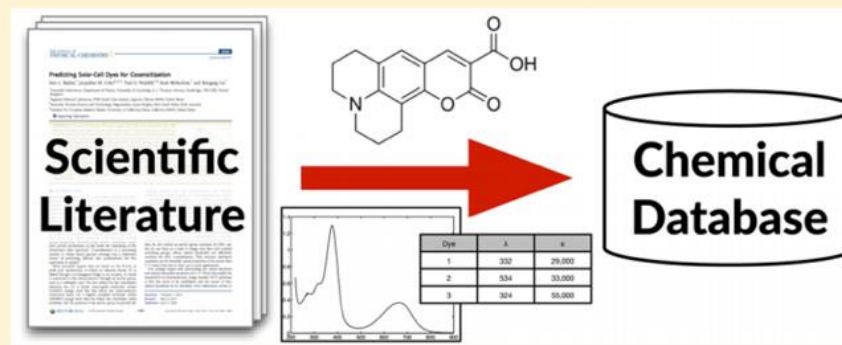
- chemistry-aware natural language processing pipeline
 - trained by unsupervised word clustering in a “massive corpus of chemistry articles”
- document-level processing to resolve data interdependencies

ChemDataExtractor: A Toolkit for Automated Extraction of Chemical Information from the Scientific Literature

Matthew C. Swain and Jacqueline M. Cole*

Cavendish Laboratory, University of Cambridge, J. J. Thomson Avenue, Cambridge, CB3 0HE, U.K.

“(A) toolkit for the automated extraction of chemical entities and their associated properties, measurements, and relationships from scientific documents that can be used to populate structured chemical databases.”



Characteristics of note;

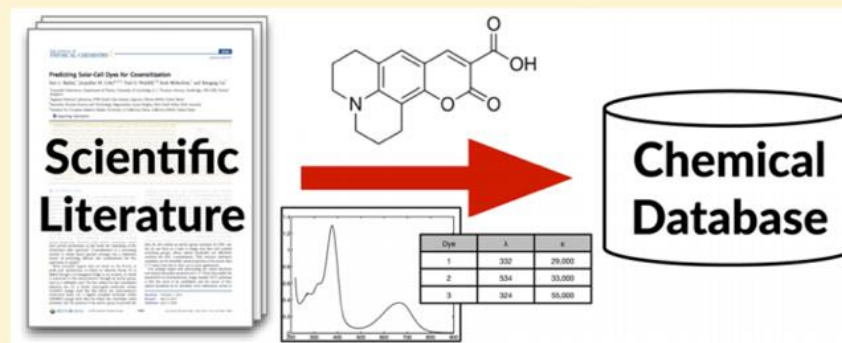
- chemistry-aware natural language processing pipeline
 - trained by unsupervised word clustering in a “massive corpus of chemistry articles”
- document-level processing to resolve data interdependencies
- The performance of the toolkit to correctly extract various types of data was evaluated

ChemDataExtractor: A Toolkit for Automated Extraction of Chemical Information from the Scientific Literature

Matthew C. Swain and Jacqueline M. Cole*

Cavendish Laboratory, University of Cambridge, J. J. Thomson Avenue, Cambridge, CB3 0HE, U.K.

“(A) toolkit for the automated extraction of chemical entities and their associated properties, measurements, and relationships from scientific documents that can be used to populate structured chemical databases.”



Characteristics of note;

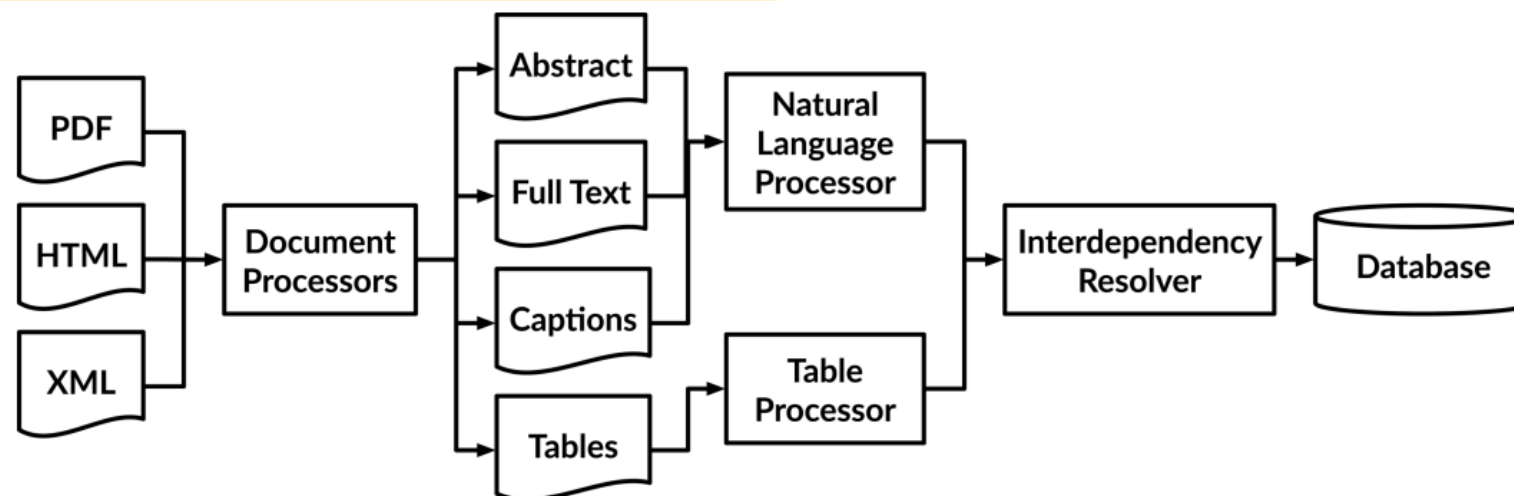
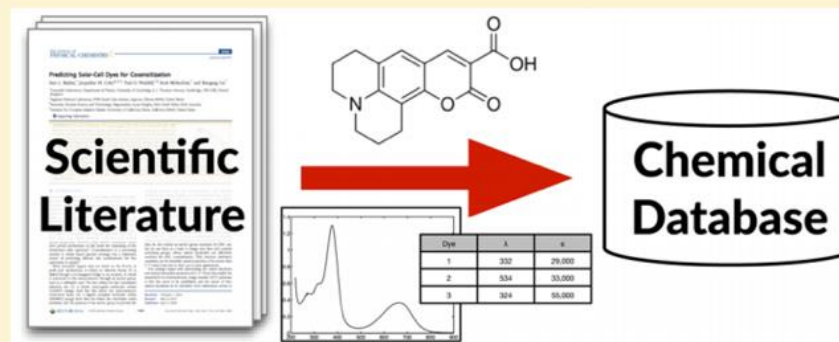
- chemistry-aware natural language processing pipeline
 - trained by unsupervised word clustering in a “massive corpus of chemistry articles”
- document-level processing to resolve data interdependencies
- The performance of the toolkit to correctly extract various types of data was evaluated
- released under the MIT license and are available to download

ChemDataExtractor: A Toolkit for Automated Extraction of Chemical Information from the Scientific Literature

Matthew C. Swain and Jacqueline M. Cole*

Cavendish Laboratory, University of Cambridge, J. J. Thomson Avenue, Cambridge, CB3 0HE, U.K.

“(A) toolkit for the **automated extraction of chemical entities and their associated properties, measurements, and relationships from scientific documents** that can be used to populate structured chemical databases.”



Output from processed document

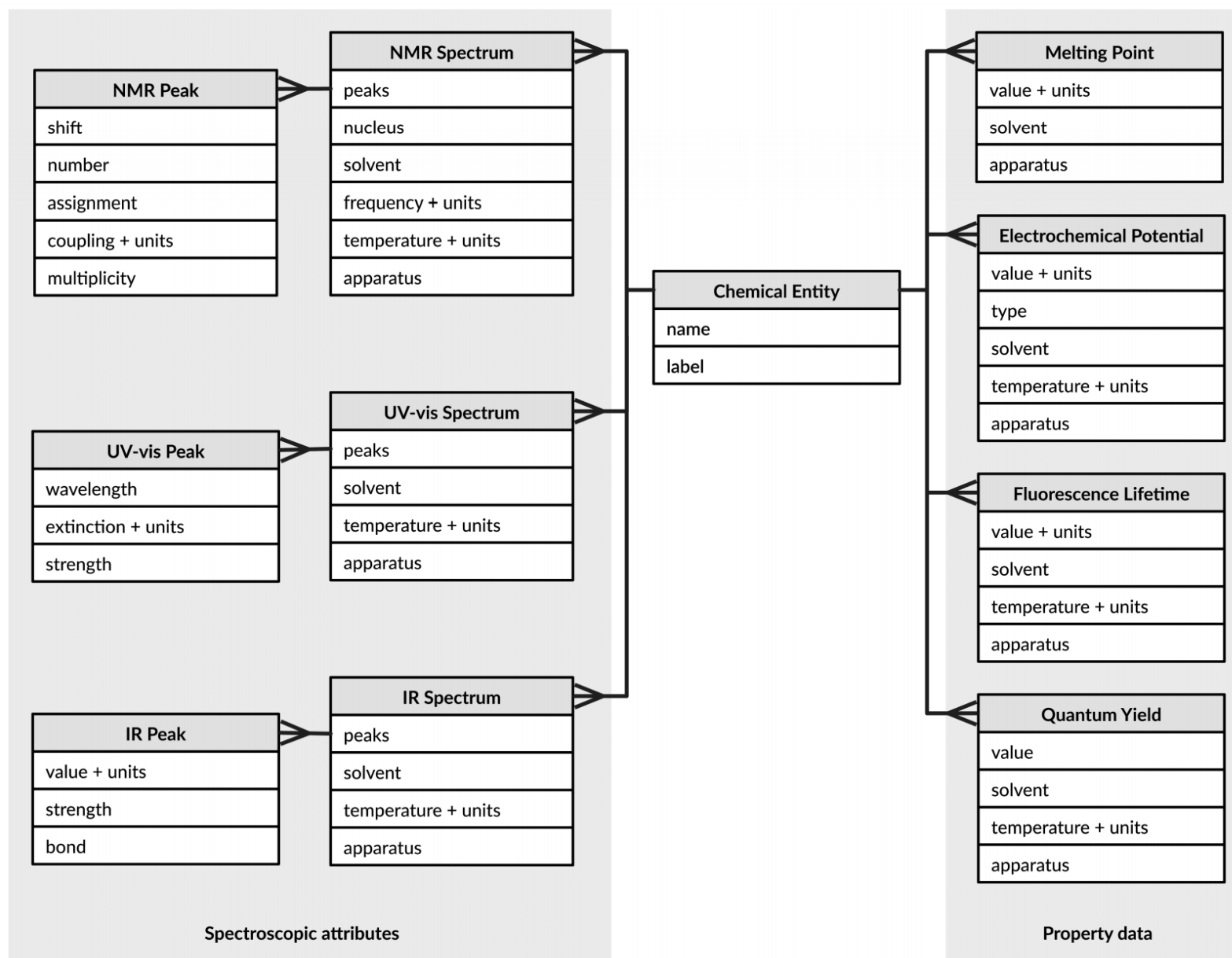


Figure 8. Data model for extracted chemical entities and their associated experimental properties and spectroscopic attributes, as currently provided by ChemDataExtractor. Users of the toolkit may extend this data model by defining their own custom parsers.

ChemDataExtractor Documentation

ChemDataExtractor is a toolkit for automatically extracting chemical information from scientific documents.

This guide provides a quick tour through ChemDataExtractor concepts and functionality.

Features

- HTML, XML and PDF document readers
- Chemistry-aware natural language processing pipeline
- Chemical named entity recognition
- Rule-based parsing grammars for property and spectra extraction
- Table parser for extracting tabulated data
- Document processing to resolve data interdependencies

Citing

If you use ChemDataExtractor as a resource in your research, please cite the following work:

Swain, M. C., & Cole, J. M. "ChemDataExtractor: A Toolkit for Automated Extraction of Chemical Information from the Scientific Literature", J. Chem. Inf. Model. **2016**, 56 (10), pp 1894–1904 [10.1021/acs.jcim.6b00207](https://doi.org/10.1021/acs.jcim.6b00207)

This project has been funded by the UK Engineering and Physical Sciences Research Council (EPSRC).

Table of Contents

[Introduction](#)
[Installation](#)
[Getting Started](#)
[Reading Documents](#)
[Chemical Records](#)
[Tokenization](#)
[Part-of-speech Tagging](#)
[Chemical Named Entities](#)
[Lexicon](#)
[Abbreviation Detection](#)
[Command Line Interface](#)
[Scraping Structured Data](#)
[Contributing](#)

GitHub, Inc. [US] | https://github.com/mcs07/ChemDataExtractor

Search or jump to... Pull requests Issues Marketplace Explore

mcs07 / ChemDataExtractor Watch 12 Star 65 Fork 25

Code Issues 3 Pull requests 4 Projects 0 Wiki Insights

Automatically extract chemical information from scientific documents <http://chemdataextractor.org>

information-extraction python chemistry text-mining natural-language-processing nlp

106 commits 1 branch 7 releases 2 contributors MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

mcs07 appveyor: Try 32 bit miniconda for python 2.7 Latest commit 349a3be on Feb 21, 2017

chemdataextractor	Fix stdin/stdout encoding issues	2 years ago
examples	Tidy up jupyter notebook example	2 years ago
requirements	Add dev requirements	2 years ago
scripts	Remove unnecessary file header information	2 years ago
tests	Remove unnecessary file header information	2 years ago
.bumpversion.cfg	Bump version: 1.2.3 → 1.3.0	2 years ago
.gitignore	Initial commit	3 years ago
.travis.yml	travis: Add python 3.6 and drop python 3.4	2 years ago
CHANGELOG.md	Add changelog	2 years ago
CONTRIBUTING.md	Add contributing guide	2 years ago
LICENSE	Remove unnecessary file header information	2 years ago
MANIFEST.in	Include more files in source distribution	2 years ago
README.rst	Remove unnecessary file header information	2 years ago
appveyor.yml	appveyor: Try 32 bit miniconda for python 2.7	2 years ago
requirements.txt	Update requirements	2 years ago
setup.py	Bump version: 1.2.3 → 1.3.0	2 years ago

Sounds great! But should we use it?

- Clean GitHub repo.
 - Issues: 3 open / 6 closed. stagnant since Jan 2018
 - Pull requests: 4 open / 10 closed.

GitHub repository page for **mcs07 / ChemDataExtractor**. The repository is a Python package for extracting chemical information from scientific documents. It includes a README, a requirements file, and a setup file. The repository has 106 commits, 1 branch, 7 releases, and 2 contributors.

Files and folders in the repository:

- chemdataextractor: Fix stdin/stdout encoding issues (2 years ago)
- examples: Tidy up jupyter notebook example (2 years ago)
- requirements: Add dev requirements (2 years ago)
- scripts: Remove unnecessary file header information (2 years ago)
- tests: Remove unnecessary file header information (2 years ago)
- .bumpversion.cfg: Bump version: 1.2.3 → 1.3.0 (2 years ago)
- .gitignore: Initial commit (3 years ago)
- .travis.yml: travis: Add python 3.6 and drop python 3.4 (2 years ago)
- CHANGELOG.md: Add changelog (2 years ago)
- CONTRIBUTING.md: Add contributing guide (2 years ago)
- LICENSE: Remove unnecessary file header information (2 years ago)
- MANIFEST.in: Include more files in source distribution (2 years ago)
- README.rst: Remove unnecessary file header information (2 years ago)
- appveyor.yml: appveyor: Try 32 bit miniconda for python 2.7 (2 years ago)
- requirements.txt: Update requirements (2 years ago)
- setup.py: Bump version: 1.2.3 → 1.3.0 (2 years ago)

Sounds great! But should we use it?

- Clean GitHub repo.
 - Issues: 3 open / 6 closed. stagnant since Jan 2018
 - Pull requests: 4 open / 10 closed.
- Written in python, compatible with python 3
- Jupyter notebook examples

266 lines (265 sloc) | 5.79 KB

Extracting a Custom Property ¶

```
In [1]: from chemdataextractor import Document
        from chemdataextractor.model import Compound
        from chemdataextractor.doc import Paragraph, Heading
```

Example Document

Let's create a simple example document with a single heading followed by a single paragraph:

```
In [2]: d = Document(
        Heading(u'Synthesis of 2,4,6-trinitrotoluene (3a)'),
        Paragraph(u'The procedure was followed to yield a pale yellow solid (b.p. 240 °C)')
        )
```

Performance

- F-scores: 93.4% (chemical identifier extraction), 86.8% (spectroscopic attribute extraction), and 91.5% (chemical property attributes)
 - CHEMDNER chemical name extraction challenge: F-score of 87.8%, vs. high scores of 87-88%