Agatha Documentation

Release 0.1.0

Carl Simon Adorf, Wenbo Shen

June 10, 2016

Contents

Agatha is a machine-learning powered literature management tool.

The tool allows you to train a *scoring model* for an existing evaluated literature database. This allows you to quickly *pre-score* other literature based on the same model.

CHAPTER 1

Installation

Agatha requires python 3 and can be installed via:

```
$ python setup.py install
```

Please see the `requirements.txt` file for external package dependencies.

CHAPTER 2

Quickstart

For the purpose of **Agatha**, all literature is identified by a unified resource identifier (URI). Common examples for URIs used in this context are `doi://10.1000/xyz123`, `arxiv://1501.0001`, or `http://www.example.com`.

For the purpose of creating a training set, create a file containing a list of URIs and a score value, e.g.:

```
# input.txt
doi://10.1000/xyz123 0.2
doi://10.1010/abc456 0
arxiv://1501.001 0.3
http://www.example.com 0.2
# and so on
```

The score is a value between 0 and 1, where 0 means *not relevant* and 1 means *highly relevant*. Training for *not relevant* entries actually improves the model!

Agatha can help you to create a training set, see *training*.

Next, we train the model and store it in a file called `model.json`.

```
agatha train input.txt > model.json
```

Finally we can take a different set of data and score it with this model:

```
agatha score --model model.json literature.txt >> scored.txt
```

The `scored.txt` will contain the URIs and the score value based on the model sorted by score. The `literature.txt` file has the same format as the `input.txt` file, with the only difference that the score-value may be missing.

CHAPTER 3

Training

There are two primary ways of creating a training set: *batch* and *interactively*.

The *batch* method is simply appending the same score value to a input set which is useful when you have pre-categorized libraries.

```
$ bib2uri mylib.bib > mylib.txt
$ agatha score mylib.txt -s 0.8 >> train.txt
```

Here, we are using the `bib2uri` script, which is automatically installed with **Agatha** to extract the URIs from the BibTeX library file.

You can use **Agatha** *interactively* to go through a new list of unrated resources. **Agatha** will attempt to obtain as many information about the specified resource and ask you to rate it on a scale from 0 to 5.

```
$ agatha score input.txt --ignore train.txt >> train.txt


Scoring 'doi://10.1103/PhysRevLett.70.2924':
Journal: Phys. Rev. Lett.
Title: Formation of a dodecagonal quasicrystalline phase in a simple
monatomic liquid
Authors: Dzugutov, Mikhail
Keywords: None
Abstract: In a recent paper M. Dzugutov, Phys. Rev. Lett. 70 2924
(1993), describes a molecular dynamics cooling simulation where he
obtained a large monatomic dodecagonal quasicrystal from a melt. The
structure was stabilized by a special potential [Phys. Rev. A46 R2984
(1992)] designed to prevent the nucleation of simple dense crystal
structures. In this comment we will give evidence that the ground
state structure for Dzugutov's potential is an ordinary bcc crystal.
Enter score [s|1..5] (s):
```

You can either enter a score or 's' for *skip*. If you just hit enter, the default value will be used which is either *skip* or the value already provided by the input set.

We are using the `-i/--ignore` argument to skip all resources that are already in the output set. This enables us to stop and restart the scoring process at any point in time.

Note: The interactive scoring scale ranges from 0 to 5 instead of 0 to 1 to allow fast and intuitive *5-star-rating*. The value is always normalized before storing.

CHAPTER 4

Sources

To fetch new literature and obtain resource information, **Agatha** supports the configuration of *sources*. To configure a *source*, simply create a `agatha_config.py` file in your home directory. For example, if you would like to specify all BibTeX-files in your home directory as a source, you would create a config file like this:

```
# ~/agatha_config.py
from agatha.sources import BibTexLibrary

sources = [
  BibTexLibrary('~/*.bib')  # wild-cards are allowed!
  ]
```

CHAPTER 5

Indices and tables

- genindex
- modindex
- search