negbio Documentation

Release 1.0

Yifan Peng

Oct 25, 2019

Contents:

CHAPTER 1

Getting Started with NegBio

These instructions will get you a copy of the project up and run on your local machine for development and testing purposes. The package should successfully install on Linux (and possibly macOS).

1.1 Installing

1.1.1 Prerequisites

- python >2.4
- Linux
- Java

Note: since v1.0, MetaMap is not required. You can use the CheXpert vocabularies (`negbio/chexpert/phrases`) instead. If you want to use MetaMap, it can be downloaded from https://metamap.nlm.nih.gov/MainDownload.shtml. Installation instructions can be found at https://metamap.nlm.nih.gov/Installation.shtml. Please make sure that both `skrmedpostctl` and `wsdserverctl` are started.

1.1.2 Installing from source (recommended)

```
$ git clone https://github.com/ncbi-nlp/NegBio.git
$ cd /path/to/negbio
$ python setup.py install --user
$ export PATH=~/.local/bin:$PATH
```

1.1.3 Installing from pip

```
$ pip install negbio
```

1.2 Using NegBio

1.2.1 Prepare the dataset

The inputs can be in either plain text or BioC format. If the reports are in plain text, each report needs to be in a single file. Some examples can be found in the `examples` folder.

1.2.2 Run the script

There are two ways to run the pipeline.

1.2.2.1 Using CheXpert algorithm

If you want to use the CheXpert method, run one of the following lines

```
$ main_chexpert text --output=examples/test.neg.xml examples/00000086.txt examples/
↪00019248.txt
```

```
$ main_chexpert bioc --output=examples/test.neg.xml examples/1.xml
```

The script will

1. [Optional] Combine `examples/00000086.txt` and `examples/00019248.txt` into one BioC XML file
2. Detect concepts using CheXpert pre-defined vocabularies (by default using the list `negbio/chexpert/phrases`)
3. Detect positive, negative and uncertain concepts using rules in `negbio/chexpert/patterns`
4. Save the results in `examples/test.neg.xml`

More options (e.g., setting the CUI list or rules) can be obtained by running

```
$ main_chexpert --help
```

1.2.2.2 Using MetaMap

If you want to use MetaMap, run the following command by replacing `<METAMAP_BIN>` with the actual **ABSOLUTE** path, such as **META_MAP_HOME/bin/metamap16**

```
$ export METAMAP_BIN=META_MAP_HOME/bin/metamap16
$ main_mm text --metamap=$METAMAP_BIN --output=examples/test.neg.xml \
    examples/00000086.txt examples/00019248.txt
```

```
$ export METAMAP_BIN=META_MAP_HOME/bin/metamap16
$ main_mm bioc --metamap=$METAMAP_BIN --output=examples/test.neg.xml examples/1.xml
```

The script will

1. [Optional] Combine `examples/00000086.txt` and `examples/00019248.txt` into one BioC XML file
2. Detect UMLS concepts (CUIs) using MetaMap (by default using the CUI list `examples/cuis-cvpr2017.txt`

3. Detect negative and uncertain CUIs using rules in `negbio/patterns`
4. Save the results in `examples/test.neg.xml`

More options (e.g., setting the CUI list or rules) can be obtained by running

```
$ main_mm --help
```

1.3 Next Steps

To start learning how to use NegBio, see the *NegBio User Guide*.

CHAPTER 2

NegBio User Guide

2.1 Run the pipeline step-by-step

The step-by-step pipeline generates all intermediate documents. You can easily rerun one step if it makes errors. The whole steps are

1. `text2bioc` combines text into a BioC XML file.
2. `normalize` removes noisy text such as `[**Patterns**]`.
3. `section_split` splits the report into sections based on titles at `patterns/section_titles.txt`
4. `ssplit` splits text into sentences.
5. Named entity recognition
    a. `dner_mm` detects UMLS concepts using MetaMap.
    b. `dner_chexpert` detects concepts using the CheXpert vocabularies at `negbio/chexpert/phrases`.
6. `parse` parses sentence using the Bllip parser.
7. `ptb2ud` converts the parse tree to universal dependencies using Stanford converter.
8. Negation detection
    a. `neg` detects negative and uncertain findings.
    b. `neg_chexpert` detects positive, negative and uncertain findings (recommended)
9. `cleanup` removes intermediate information.

Steps 2-10 will process the input files one-by-one and generate the results in the output directory. The 2nd and 3rd can be skipped. You can chose either step 5 or 6 for named entity recognition.

2.1.1 1. Convert text files to BioC format

You can skip this step if the reports are already in the BioC format. **If you have lots of reports, it is recommended to put them into several BioC files, for example, 100 reports per BioC file.**

```
$ export BIOC_DIR=/path/to/bioc
$ export TEXT_DIR=/path/to/text
$ negbio_pipeline text2bioc --output=$BIOC_DIR/test.xml $TEXT_DIR/*.txt
```

Another most commonly used command is:

```
$ find $TEXT_DIR -type f | negbio_pipeline text2bioc --output=$BIOC_DIR
```

2.1.2 2. Normalize reports

This step removes the noisy text such as `[**Patterns**]` in the MIMIC-III reports.

```
$ negbio_pipeline normalize --output=$OUTPUT_DIR $INPUT_DIR/*.xml
```

2.1.3 3. Split each report into sections

This step splits the report into sections. The default section titles is at `patterns/section_titles.txt`. You can specify customized section titles using the option `--pattern=<file>`.

```
$ negbio_pipeline section_split --output=$OUTPUT_DIR $INPUT_DIR/*.xml
```

2.1.4 4. Splits each report into sentences

This step splits the report into sentences using the NLTK splitter (nltk.tokenize.sent_tokenize).

```
$ negbio_pipeline ssplit --output=$OUTPUT_DIR $INPUT_DIR/*.xml
```

2.1.5 5. Named entity recognition

This step recognizes named entities (e.g., findings, diseases, devices) from the reports. The first version of NegBio uses MetaMap to detect UMLS concepts.

MetaMap can be can be downloaded from https://metamap.nlm.nih.gov/MainDownload.shtml. Installation instructions can be found at https://metamap.nlm.nih.gov/Installation.shtml. Before using MetaMap, please make sure that both `skrmedpostctl` and `wsdserverctl` are started.

MetaMap intends to extract all UMLS concepts. Many of them are not irrelevant to radiology. Therefore, it is better to specify the UMLS concepts of interest via `--cuis=<file>`

```
$ export METAMAP_BIN=META_MAP_HOME/bin/metamap16
$ negbio_pipeline dner_mm --metamap=$METAMAP_BIN --output=$OUTPUT_DIR $INPUT_DIR/*.xml
```

NegBio also integrates the CheXpert vocabularies to recognize the presence of 14 observations. All vocabularies can be found at `negbio/chexpert/phrases`. Each file in the folder represents one type of named entities with various text expressions. So far, NegBio does not support adding more types in the folder, but you can add more text expressions of the type.

```
$ negbio_pipeline dner_chexpert --output=$OUTPUT_DIR $INPUT_DIR/*.xml
```

In general, MetaMap is more comprehensive while CheXpert is more accurate on 14 types of findings. MetaMap is also slower and easier to break than CheXpert.

2.1.6 6. Parse the sentence

This step parses sentence using the Bllip parser.

```
$ negbio_pipeline parse --output=$OUTPUT_DIR $INPUT_DIR/*.xml
```

2.1.7 7. Convert the parse tree to UD

This step converts the parse tree to universal dependencies using Stanford converter.

```
$ negbio_pipeline ptb2ud --output=$OUTPUT_DIR $INPUT_DIR/*.xml
```

2.1.8 8. Detect negative and uncertain findings

This step detects negative and uncertain findings using patterns. By default, the program uses the negation and uncertainty patterns in the `negbio/patterns` folder. However, you are free to create your own patterns via `--neg-patterns=<file>` and `--uncertainty-patterns=<file>`. The pattern is a semgrex-type pattern for matching node in the dependency graph. Currently, we only support < and > operations. A detailed grammar specification (using PLY, Python Lex-Yacc) can be found in `ngrex/parser.py`.

```
$ negbio_pipeline neg --output=$OUTPUT_DIR $INPUT_DIR/*.xml
```

NegBio also integrates the CheXpert algorithms. Different from the original NegBio, CheXpert utilizes a 3-phase pipeline consisting of pre-negation uncertainty, negation, and post-negation uncertainty (Irvin et al., 2019). Each phase consists of rules which are matched against the mention; if a match is found, then the mention is classified accordingly (as uncertain in the first or third phase, and as negative in the second phase). If a mention is not matched in any of the phases, it is classified as positive.

Generally, the CheXpert contains more rules and is more accurate than the original NegBio.

```
$ negbio_pipeline neg_chexpert --output=$OUTPUT_DIR $INPUT_DIR/*.xml
```

Similarly, you are free to create patterns via `--neg-patterns=<file>`, `--pre-uncertainty-patterns=<file>`, and `--post-uncertainty-patterns=<file>`.

2.1.9 9. Cleans intermediate information

This step removes intermediate information (sentence annotations) from the BioC files.

```
$ negbio_pipeline cleanup --output=$OUTPUT_DIR $INPUT_DIR/*.xml
```

CHAPTER 3

NegBio Developer Guide

3.1 Create the documentation

Install Sphinx

```
$ pip install Sphinx
$ pip install sphinx_rtd_theme
$ cd docs
$ make html
```

CHAPTER 4

License

PUBLIC DOMAIN NOTICE

National Center for Biotechnology Information

This software/database is a "United States Government Work" under the terms of the United States Copyright Act. It was written as part of the author's official duties as a United States Government employee and thus cannot be copyrighted. This software/database is freely available to the public for use. The National Library of Medicine and the U.S. Government have not placed any restriction on its use or reproduction.

Although all reasonable efforts have been taken to ensure the accuracy and reliability of the software and data, the NLM and the U.S. Government do not and cannot warrant the performance or results that may be obtained by using this software or data. The NLM and the U.S. Government disclaim all warranties, express or implied, including warranties of performance, merchantability or fitness for any particular purpose.

Please cite the author in any work or product based on these materials:

Peng Y, Wang X, Lu L, Bagheri M, Summers RM, Lu Z. NegBio: a high-performance tool for negation and uncertainty detection in radiology reports. AMIA 2018 Informatics Summit. 2018.

Wang X, Peng Y, Lu L, Bagheri M, Lu Z, Summers R. ChestX-ray8: Hospital-scale Chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017, 2097-2106.

CHAPTER 5

Contributing

Please read `CONTRIBUTING.md` for details on our code of conduct, and the process for submitting pull requests to us.

CHAPTER 6

Acknowledgments

This work was supported by the Intramural Research Programs of the National Institutes of Health, National Library of Medicine and Clinical Center.

We are grateful to the authors of NegEx, MetaMap, Stanford CoreNLP, Bllip parser, and CheXpert labeler for making their software tools publicly available.

We thank Dr. Alexis Allot for the helpful discussion.

CHAPTER 7

Disclaimer

This tool shows the results of research conducted in the Computational Biology Branch, NCBI. The information produced on this website is not intended for direct diagnostic use or medical decision-making without review and oversight by a clinical professional. Individuals should not change their health behavior solely on the basis of information produced on this website. NIH does not independently verify the validity or utility of the information produced by this tool. If you have questions about the information produced on this website, please see a health care professional. More information about NCBI's disclaimer policy is available.

CHAPTER 8

Reference

- Peng Y, Wang X, Lu L, Bagheri M, Summers RM, Lu Z. NegBio: a high-performance tool for negation and uncertainty detection in radiology reports. *AMIA 2018 Informatics Summit*. 2018.
- Wang X, Peng Y, Lu L, Bagheri M, Lu Z, Summers R. ChestX-ray8: Hospital-scale Chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, 2097-2106.

CHAPTER 9

Indices and tables

- genindex
- modindex
- search