Semi-Open Relation Extraction (SORE)

Release 0.1

Ruben Kruiper

Jun 06, 2020

CONTENTS

Auto-generated documentation for the SORE code found in the FOBIE repo: https://github.com/rubenkruiper/FOBIE

- genindex
- modindex

If you have any questions, please let me know!

CHAPTER

ONE

READ ME

To test SORE this repository contains ~110 papers, which are copied from the [OA-STM corpus](https://github.com/elsevierlabs/OA-STM-Corpus). These papers are sourced from 10 different scientific domains: Agriculture, Astronomy, Biology, Chemistry, Computer Science, Earth Science, Engineering, Materials Science, Mathematics and Medicine.

1.1 Train and use SciIE

Clone the SciIE code from [this bitbucket repository](https://bitbucket.org/luanyi/scierc/src/master/), also check this repository for more information on adjusting runtime parameters.

Create a separate SciIE environment:

- *conda create –name SCIIE python=2.7*
- *source activate SCIIE*

Install the following dependencies:

- *pip install TensorFlow==1.8.0*
- *pip install tensorflow_hub*
- *pip install pyhocon*

1. Fetch GloVe embeddings and build kernels

   Open the SciIE folder in a terminal/console window and run:

   - *./scripts/fetch_required_data.sh*
   - *./scripts/build_custom_kernels.sh*

2. Generate ELMo embeddings for FOBIE and train a model

   First copy the SciIE formatted FOBIE files to: *[SciIE_folder]/data/processed_data/json/*

   In the SORE directory you can find a folder called *sciie_scripts/*, copy the files to your *[SciIE_folder]*:

   - Replace *[SciIE_folder]/experiments.conf* and place *embed_and_predict.py* and *generate_FOBIE_embeddings.py* alongside it
   - Create a directory *[SciIE_folder]/FOBIE_output/* to store predictions

   Generate the embeddings for the FOBIE dataset:

   - Run *python generate_FOBIE_embeddings.py* (make sure to be in the correct environment)

   To train a model you can run:

- *python singleton.py fobie_train & python evaluator.py fobie_train*

3. Predict using your trained model

   You're now set to make predictions on unseen data:

   - *python embed_and_predict.py fobie_train*

   This will loop through the files found in *[SciIE_folder]/data/processed_data/json/* and ask you which files to process before starting the embedding (can take a while for big files).

1.2 Prepare OpenIE 5

Clone the [github repo for OpenIE 5](https://github.com/dair-iitd/OpenIE-standalone), aslo see this repo for more information about compiling. In the newly created *OpenIE-standalone* folder, you'll have to create a new directory called *lib/*.

You will have to download and place into the *lib/* folder:

- [The BONIE standalone jar](https://github.com/dair-iitd/OpenIE-standalone/releases/download/v5.0/BONIE.jar)
- [The CALMIE standalone jar](https://github.com/dair-iitd/OpenIE-standalone/releases/download/v5.0/ListExtractor.jar)

Furthermore, you will have to download and place into the *data/* folder:

- [The Berkeley Language Model](https://drive.google.com/file/d/0B-5EkZMOlIt2cFdjYUJZdGxSREU/view?usp=sharing)

OpenIE is compiled using *sbt* and Java 8:

- Install [SDK](https://sdkman.io/install) by running *curl -s "https://get.sdkman.io" | bash*
- Open a new terminal window and check that sdk is installed *sdk version*
- Then install sbt *sdk install sbt*: * You may need sbt at version 0.13.x , e.g.: *sdk install sbt 0.13.18*
- Inside the *OpenIE-standalone/* directory: * First run *bash compile.sh* to compile SRLIE and ONRE * Then compile the jar file *sbt -J-Xmx10000M clean compile assembly* * You may have to [install Java 8](https://www.scala-sbt.org/1.x/docs/Installing-sbt-on-Mac.html) first , e.g.: *sdk install java 8.0.252-amzn* * And add java to your path, e.g., *export PATH=$PATH:~/.sdkman/candidates/java/8.0.252-amzn/bin* * You may need to set the Scala version to [version 2.10.2](https://www.scala-lang.org/download/2.10.2.html)

1.3 Run SORE

Create an environment and activate, e.g.;

- *conda create –name SORE python=3.6*
- *source activate SORE*

Install the requirements found inside *[FOBIE_repo]/SORE/*:

- *pip install -r requirements.txt*
- Download required language models * *python -m spacy download en_core_web_sm* * *python -m textblob.download_corpora*

Edit the *SORE_settings.json* doc:

- Set the *path_to_OIE_jar*, for example: */Users/../../OpenIE-standalone/target/scala-2.10/openie-assembly-5.0-SNAPSHOT.jar*

You can run the code for SORE from the FOBIE repository:

- *python run_SORE.py*

Main settings:

- Prepare_data: Yes/No - convert json data found in *[FOBIE_repo]/SORE/data/unprocessed/* to OpenIE and SciIE input files. * Note that to predict with SciIE you'll have to move these input files to *[SciIE_folder]/data/processed_data/json/*.
- Parse_narrowIE_predictions: Yes/No - parse SciIE prediction files to a single csv, specified in the settings file under *"narrowIE_input_files"*. These predictions files should be placed in *[FOBIE_repo]/SORE/data/narrowIE/predictions/*, note that you'll have to move predictions made with SciIE here (from the *[SciIE_folder]/FOBIE_output/* folder).
- Run_OIE: Yes/No - run OpenIE 5 on each of the files for which SciIE has found relations.
- Filter_OIE: Yes/No - filter the OpenIE 5 extractions using the predicted relations stored in the narrowIE output csv file.
- convert_back_to_BRAT: Yes/No - convert the SORE extractions to a BRAT annotations format, this enables you to visualise the resulting extractions in [BRAT](https://brat.nlplab.org/index.html).

Note that FOBIE focuses on tradeoffs and other types of correlations in the Biology domain. Non-tradeoff relations can be used to filter OpenIE extractions in distinct domains, but whether the arguments used for filtering are useful can vary.

If you have any questions, please let me know!

CHAPTER

TWO

SORE DOCUMENTATION

2.1 run_SORE.py main()

2.2 SORE package

2.2.1 Prepare input data

2.2.2 Parse narrow IE predictions

2.2.3 Compute OpenIE 5 extractions

2.2.4 SORE filtering

2.2.5 Convert SORE output to BRAT annotations

2.2.6 Module contents

2.3 SORE utilities

2.3.1 IDF weight utils

2.3.2 Clean input strings

2.3.3 Convert json to OpenIE5 input format

2.3.4 Convert json to SciIE input format

2.3.5 SORE filtering utilities

2.3.6 Custom spacy pipeline

2.3.7 Module contents