README

David Byrne, Robert Goodhead, Michael McMahon, Conor Parle

This is a readme file, designed to allow researchers to apply the approach of Byrne et al. (2022a). The files here represent a minimal working example (MWE) and will parse 9 text files from containing policy statements by the Federal Reserve in 2019.

# Preamble

You will need the following programmes on your system, and accessible from your system path: Python, Java JDK, Maeven, Git Bash.

# Step 1: Ensure the file structure is correct

The file structure for the working directory needs to be as follows:

* ./data
* ./mate\_tools\_working
* ./stanford-corenlp-4.0.0
* ./tmv\_tool

The file structure for ./data/data\_MWE needs to be as follows:

* ./data/data\_MWE
  + /corpus
  + /corpus\_prepared\_for\_mate
  + /mate\_parsed\_corpus
  + /mate\_parsed\_corpus\_backup
  + /sent\_corpus
  + /sent\_dates\_corpus
  + /sent\_sutime\_corpus
  + /tmv

The correct file structures for ./mate\_tools\_working, ./stanford-corenlp-4.0.0, and ./tmv\_tool will be discussed subsequently.

Ensure the folder ./data/data\_MWE/corpus includes the 9 text files of interest. However, any text file containing English language sentences, will be parsed if they are placed in this folder. It is necessary that they follow the naming convention “filename\_YYYYMMDD.txt” however, where YYYYMMDD indicates the reference date of the document.

# Step 2: Pre-process the textual data

You will need to run the following python codes in sequence to pre-process the data:

* run\_MWE\_1a\_preprocessing\_minimal.py
* run\_MWE\_1c\_preprocessing\_for\_mate.py
* run\_MWE\_1ca\_dates\_preprocessing\_for\_sutime.py

# Step 3: Set up the TMV tool

Before one can run the TMV tool, one needs to set up the MATE parser.

## 3a) Set up MATE

* Download the file anna-3.61.jar from [here](https://code.google.com/archive/p/mate-tools/downloads) and put it in a folder called ./mate\_tools\_working/anna.
* Download transition-1.30.jar from [here](https://code.google.com/archive/p/mate-tools/wikis/ParserAndModels.wiki) and put it in a folder called ./mate\_tools\_working/transition.
* Download the parser + tagger from [here](https://code.google.com/archive/p/mate-tools/wikis/ParserAndModels.wiki), under “English Models”, which is a .mdl file called per-eng-S2b-40.mdl and put it in a folder called ./mate\_tools\_working/parser\_tagger/.
* Download the parser .csh script from [here](https://code.google.com/archive/p/mate-tools/wikis/ParserAndModels.wiki), which is a tiny example script called parse-eng, and put it in ./mate\_tools\_working/parser\_tagger/.

## 3b) Set up TMV

* Download the TMV tools (from the GitHub repository [here](https://github.com/aniramm/tmv-annotator)) into this folder and unzip inside ./tmv\_tool. There should be folders called europarl, example-outputs, and tmv-annotator-tool. There should also be .gitignore, LICENSE, README.md.
* Note that you need the de-bugged version of the English variant of the tool. You should place the file TMV-EN\_ecb\_test\_david.py (found in ./supplements) in the directory (./data/tmv-annotator-tool).
* You also need to add the file TMVtoHTML\_ecb.py (found in ./supplements) in the directory (./data/tmv-annotator-tool).

# Step 4: Getting SUTime working

You will need a version of Stanford CoreNLP on your system. In our applications, we used version 4.0.0, which is available [here](https://stanfordnlp.github.io/CoreNLP/history.html). While our codes may function with more recent versions, we cannot guarantee this will be the case.

The folder ./stanford-corenlp-4.0.0 should have the following file structure:

* ./stanford-corenlp-4.0.0
  + /.idea
  + /jars
  + /patterns
  + /sutime
  + /target
  + /tokensregex

Next one needs to add two additional rules files, that are bespoke to this paper and are not included in the core distribution of SUTime. To do this, move the files defs2.sutime.txt and english2.sutime.txt (found in ./supplements) to the folder ./stanford-corenlp-4.0.0/sutime.

One now needs to add a .java file, designed to run extract the reference date from text file names, before applying SUTime to this reference date. To do this move the two files run\_sutime\_on\_MWE.java and run\_sutime\_on\_MWE.class into ./stanford-corenlp-4.0.0. These two files are both found in ./supplements.

# Step 5: Parse the data

To parse the data with TMV, one needs to run the following two files in sequence:

* bash\_run\_mate\_on\_MWE.sh
* bash\_run\_tmv\_on\_MWE.sh

To parse the data with SUTime, one needs to run the following file:

* bash\_run\_sutime\_on\_MWE.sh

# Step 6: Additional preparation routines

These routines apply a few cleaning operations, as detailed in Byrne et al. (2022a). These routines are specific to our investigations, and any individual cleaning decision we leave for future researchers to remove or modify as they please.

* run\_MWE\_2a\_tmv\_1\_data\_input
* run\_MWE\_2a\_tmv\_1a\_tempoword
* run\_MWE\_2b\_sutime\_datainput
* run\_MWE\_3a\_tmv\_preparation
* run\_MWE\_3b\_sutime\_preparation

The cleaned SUTime and TMV parsed data can be respectively found stored in the following .pkl files:

* ./data/data\_MWE/data\_TMV\_cleaned.pkl
* ./data/data\_MWE/data\_SUTime\_cleaned.pkl