plovigy-NGEC-review.py documentation

Philip A. Schrodt (schrodt735@gmail.com)
Last update: 30-Nov-2022

`plovigy-NGEC-review.py` is a very tiny footprint Python program—~260 lines, no dependencies except standard libraries—distantly based on the `prodigy` program[1] that is used to review the NGEC training cases. It is keyboard based and runs in a terminal using the incredibly robust golden-oldie `curses` library with no additional dependencies.

Operating Environment

This runs from the command-line in a terminal window configured (in characters) to at least 148W x 36H. It requires a reasonably current version of Python: Current version has been developed in Python 3.11 on Mac-OS 12.2.1 but I'm pretty sure it will run in anything past Python 3.6. I've been developing in the Mac-OS `Terminal` and an earlier variant ran with no modifications on Ubuntu Linux so presumably it will also run in the `Terminal` equivalent in Windows.

To Run Program:

```
python3 plovigy-NGEC-review.py <filename> [-c <coder>] [-r <category>]
[-x <one or more strings>]
```

where the required `<filename>` is the path to the event training file to read ("source file").

> The program currently expects this to contain the string "Mk7" (this is used to generate part of the output file name)

`-c <coder>` is optional and is typically the coder initials and will be incorporated into the output file name; it defaults to "PAS"

`-r <category>` is optional and is used to set an alternative category which is coded with the single keystroke R. `<category>` can be anything: the program will give a warning if it is not a PLOVER category but does not stop; this will typically be used when multiple categories are coded (e.g. `-r COERCE,PROTEST)` but can be another. Note per command line conventions, there should not be embedded blanks in the `<category>` string.

[1] `prodigy + PLOVER = plovigy`: this is the latest in a long line of programs, originally motivated by getting something that could be used on long flights where there was no access to the internet. Back when we took long flights.

`-x <one or more strings>` is optional and sets an exclusion list that simply skips any record where the text contains an element in the list. For example

```
python3 plovigy-NGEC-review.py
Annot_training_Mk9_THTN-exper.json -x sanction "Palestinian
territories" -c EXP
```

skips any record containing the terms "sanction" or "Palestinian territories". Per the usual Unix command-line rules, if a string contains a blank, it should be enclosed in quotes, otherwise it will be treated as two distinct words. The list of terms is copied either to the next option beginning with "-" or to the end of the arguments. This is typically used when reviewing synthetic cases which tend to have redundant phrases, but also works whenever you've obtained sufficient examples of texts with a distinguishing phrase. The exclusion list is written to "plovigy.ngec.filerecs.txt"

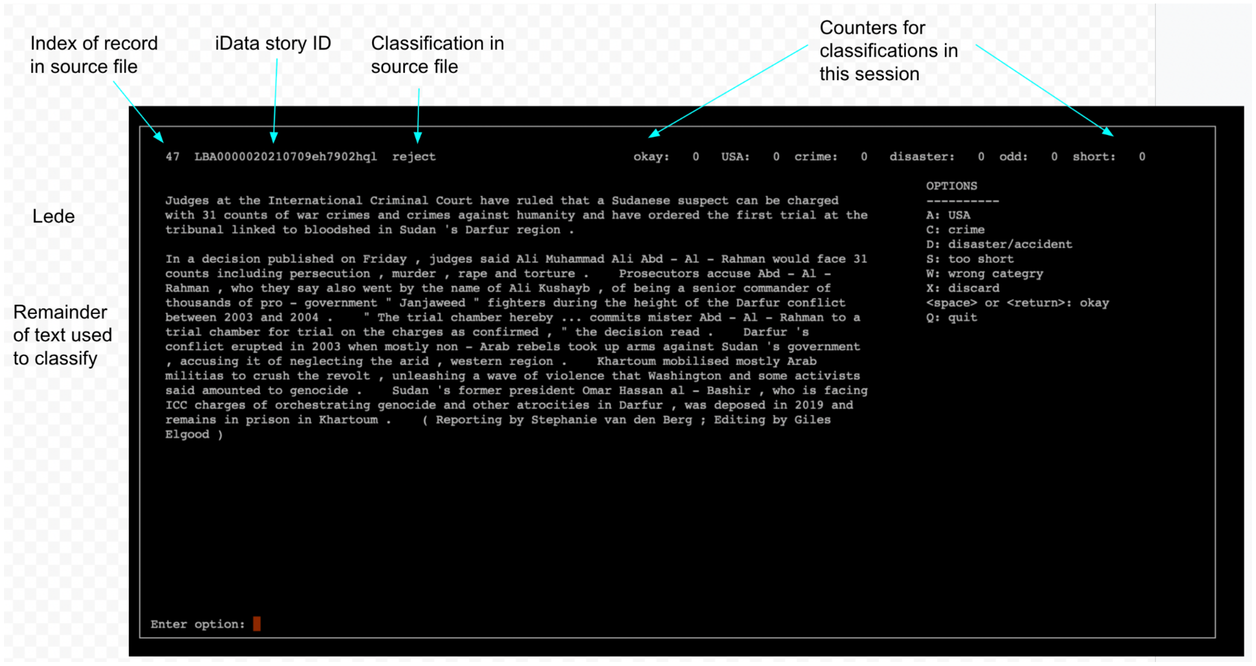
The program keeps track of the records in the source file using the file *plovigy.ngec.filerecs.txt* (this will be created if it doesn't exist). This contains entries of the form

```
Annot_training_Mk9_THTN-exper.json 5 -221216-145654.jsonlTotals:  okay:   0   USA:   0  crime:   3
disast/accid:   0   recoded:   0  odd:   0  short:   0   Excluded   3: ['sanction', 'Palestinian territories']
```

The second field ("5") gives the number of records already coded; these will be skipped the next time the program is used. An entry of "-1" indicates the program hit the end of the file and if the program is run again, it will ask whether we want to restart. This location can be changed manually in a text editor if you need to recode something or if it crashes before updating *plovigy.ngec.filerecs.txt*. The remaining fields show the counts of the various types of coding; the "Exclude" fields are only present if the -x option was used.

Coding

The main display screen is shown below and generally the program operates by simply selecting a single-key option (not case sensitive) for each case.


Index of record in source file
iData story ID
Classification in source file
Counters for classifications in this session
Lede
Remainder of text used to classify

```
47  LBA0000020210709eh7902hql  reject                     okay:   0   USA:   0  crime:   0   disaster:   0  odd:   0  short:   0

                                                                                                        OPTIONS
Judges at the International Criminal Court have ruled that a Sudanese suspect can be charged           ----------
with 31 counts of war crimes and crimes against humanity and have ordered the first trial at the       A: USA
tribunal linked to bloodshed in Sudan 's Darfur region .                                               C: crime
                                                                                                        D: disaster/accident
In a decision published on Friday , judges said Ali Muhammad Ali Abd - Al - Rahman would face 31       S: too short
counts including persecution , murder , rape and torture .    Prosecutors accuse Abd - Al -            W: wrong categry
Rahman , who they say also went by the name of Ali Kushayb , of being a senior commander of           X: discard
thousands of pro - government " Janjaweed " fighters during the height of the Darfur conflict          <space> or <return>: okay
between 2003 and 2004 .    " The trial chamber hereby ... commits mister Abd - Al - Rahman to a        Q: quit
trial chamber for trial on the charges as confirmed , " the decision read .    Darfur 's
conflict erupted in 2003 when mostly non - Arab rebels took up arms against Sudan 's government
, accusing it of neglecting the arid , western region .    Khartoum mobilised mostly Arab
militias to crush the revolt , unleashing a wave of violence that Washington and some activists
said amounted to genocide .    Sudan 's former president Omar Hassan al - Bashir , who is facing
ICC charges of orchestrating genocide and other atrocities in Darfur , was deposed in 2019 and
remains in prison in Khartoum .    ( Reporting by Stephanie van den Berg ; Editing by Giles
Elgood )



Enter option:
```

<space> or <return>: Classification is okay; go to next record

A: news internal to the USA, which we aren't coding

C: non-political criminal issue

V: both USA and crime

D: natural disaster without any international activity

F: accident without any international activity

S: record is too short. If it is really short (<`MIN_LENGTH` characters) it is automatically discarded

W: category is wrong: see discussion below

R: reset category to the string set by the `-r <category>` command-line option . R only works if this has been set; otherwise that key is ignored.

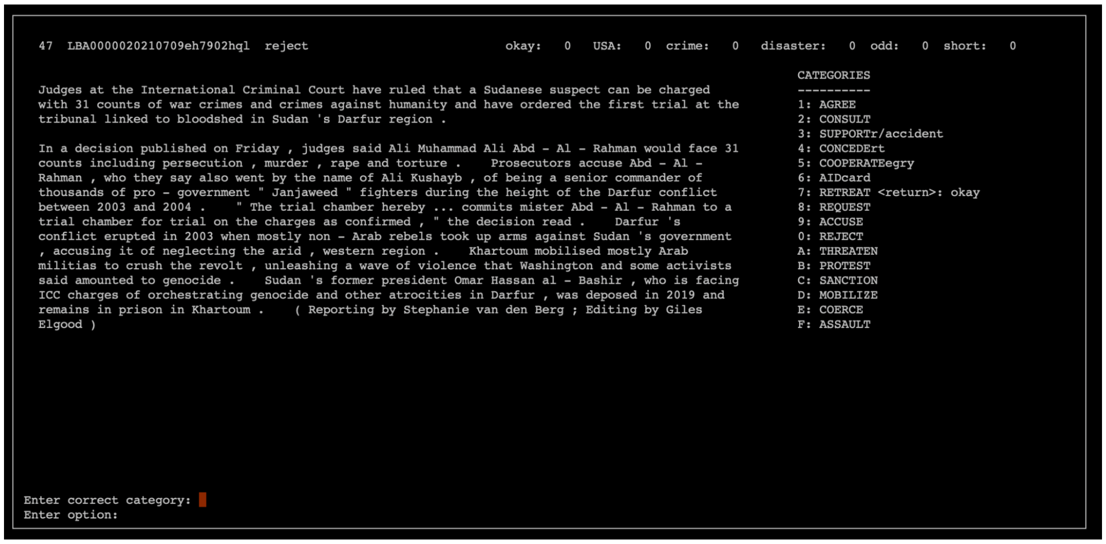
X: discard the record from the training cases for any reason

Q: quit program and update the location of the last coded record.

Recoding categories ("W")

The W key generates the screen below, with the new prompt "Enter correct categories: " A list of the accepted categories (currently the PLOVER events) is listed on the screen, and you should enter a string with the appropriate labels (e.g. "9ae" generates the categories "ACCUSE, THREATEN, COERCE"). Labels are not case sensitive; the string can be a single character; use <return> to enter the string.

If an invalid single label is in the string, you will see the message "Category X not found; try again" and returned to the "Enter correct categories: " Using the "EXIT" option will exit this loop and write the record with `"plevent": "category:EXIT"`; these records can then be edited manually (typically by just removing: presumably this only occurs by accident, though see note below on using this to code "reject")



```
  47  LBA0000020210709eh7902hql  reject                        okay:   0   USA:   0  crime:   0   disaster:   0  odd:   0  short:   0

                                                                                                            CATEGORIES
  Judges at the International Criminal Court have ruled that a Sudanese suspect can be charged              ----------
  with 31 counts of war crimes and crimes against humanity and have ordered the first trial at the          1: AGREE
  tribunal linked to bloodshed in Sudan 's Darfur region .                                                  2: CONSULT
                                                                                                            3: SUPPORTr/accident
  In a decision published on Friday , judges said Ali Muhammad Ali Abd - Al - Rahman would face 31          4: CONCEDErt
  counts including persecution , murder , rape and torture .    Prosecutors accuse Abd - Al -               5: COOPERATEegry
  Rahman , who they say also went by the name of Ali Kushayb , of being a senior commander of               6: AIDcard
  thousands of pro - government " Janjaweed " fighters during the height of the Darfur conflict             7: RETREAT <return>: okay
  between 2003 and 2004 .    " The trial chamber hereby ... commits mister Abd - Al - Rahman to a           8: REQUEST
  trial chamber for trial on the charges as confirmed , " the decision read .    Darfur 's                  9: ACCUSE
  conflict erupted in 2003 when mostly non - Arab rebels took up arms against Sudan 's government           0: REJECT
  , accusing it of neglecting the arid , western region .    Khartoum mobilised mostly Arab                 A: THREATEN
  militias to crush the revolt , unleashing a wave of violence that Washington and some activists           B: PROTEST
  said amounted to genocide .    Sudan 's former president Omar Hassan al - Bashir , who is facing          C: SANCTION
  ICC charges of orchestrating genocide and other atrocities in Darfur , was deposed in 2019 and            D: MOBILIZE
  remains in prison in Khartoum .    ( Reporting by Stephanie van den Berg ; Editing by Giles               E: COERCE
  Elgood )                                                                                                  F: ASSAULT









Enter correct category: 
Enter option:
```

Output

The records that are annotated as something other than "okay" are saved in a time-stamped JSON file which will be processed later to update the training cases. This can also be manually edited and should be okay so long as the JSON structure is preserved; the processing actually does not use the 'text' field, which is obtained from the original source, but it preserved as a processing-trail.

Some random observations

1. The operational keys are clustered really nicely…uh…provided you are left-handed and using a QWERTY keyboard…

2. I've separated out the lede (first sentence, or at least that usually works) to make the scanning easier; this distinction is not made by the classifier

3. Very rarely, the text will overflow the available screen space; it is truncated

4. There's no "reverse" at the moment; if you make a mistake record the index number (upper left corner), use that to find the record in the source file, and then copy sufficient text to find

the record in the output file (if it got written) and manually make the correction (or copy the record into the output file). Sounds complicated but goes quickly.

5. This is currently configured specifically for the event training cases, but will be generalized to work with PLOVER mode and context cases.

6. Note that in the current training configuration, negative cases are denoted by `"plevent": "reject"` which is distinct from `"plevent": "REJECT"` which is a positive case for the REJECT category. This is not particularly good practice, and could readily be refactored at some point in the future, but I'm not going to mess with it now.

   At present there is no way to change a coding to lower-case `"reject"`—that is, move it from a positive to a negative coding—as I've just been discarding (option X which generates `"plevent": "error")` cases which are miscoded but I don't see an obvious alternative code for. That's probably the best approach, though the `"plevent": "category:EXIT"` could be used for this approach, with a find-and-replace in the output file of `"category:EXIT"→"reject"`.

Changes that could be made fairly easily in the code

1. If someone wants to figure out an alternative key configuration (or add a new category similar to crime and disaster), this can be modified using the list `OPTIONLIST` and that long block of `if/elif` that interprets the keystrokes. Said block could also be fairly easily turned into a table look-up, say by changing the elements of `OPTIONLIST` to a tuple where the second element is the `plevent` code and switch the counters to a list of integers indexed on `OPTIONLIST`…"W" will still need to be handled differently…you get the idea…but we'll leave it hard-coded right now.

2. The global `NGECDIR` controls the location of the source files, it is currently just set to `./` (current directory)

3. `SUBW_HGT = 36` and `SUBW_WID = 148` control the screen size: you could increase this if working on a larger screen. You might also experiment with increasing the Terminal text size depending on your screen size and eyesight

4. `TEXT_WIDTH = 96` controls the text width; the current setting is about what I can take in without moving my eyes much

Comments on classifying

General

Generally what we are trying to do is build a set of cases that the algorithms can model, and there is a limit to the distinctions they can make. A problem I've noticed is edge-case

annotations that look *really* close to the category but aren't: in an ideal world where we had many times more training cases than we have (and a large validation set to insure these aren't causing problems), we might be able to handle these but at present we're not in that world, and are primarily interested in getting the typical examples for each category.
Also keep in mind that many times these quirky cases will be extraordinarily rare—we used to run into this problem a lot in dictionary development: you'd hit a couple examples, then never find these again (probably someone got another job…), so they are basically just introducing noise.

USA

This one is usually straightforward; note that if there is a US interaction with another international actor, we keep the case; the is just for purely internal events, which are criminal (use USA rather than crime, as the intention is to pre-filter the USA cases before they go through the classifiers), electoral or legislative.

crime

Most of the time there will be no political content—robbery, partner violence, gang violence, and the like. There's a gray area where a politician is accused of a crime (introducing the additional issue of whether the accusation has any basis: often it clearly doesn't (popular way to delegitimize the opposition and not just in Russia) but, e.g. in corruption and sexual assault cases the accusation may be credible): when in doubt leave these sorts of cases in.

disaster and accident

This applies to reports of the disaster or accident itself; if the report deals with an international *response* to a disaster or accident, it is okay

short

A lot of these are probably in there because the article started out much longer but what you see is all the remains once our current filters for boilerplate have been applied. In some cases the article is a bit longer than MIN_LENGTH but the end of it also is mostly [unfiltered] boilerplate, so it is quite unlikely to contribute to classification

wrong category

Only use this for cases classified into a category, not for "reject" cases, which are by definition in other categories. This takes more processing but given you've already read the article, might as well keep the information. These will be separated out and added to the appropriate training cases.

discard

Anything else that looks like it might mess up the classifier, but be particularly alert for variations on composite stories, e.g. the "Today's headlines…" stories: we've filtered out most of these now but there are a lot of different ways that these can be done, particularly in OSE, and they can confuse thing in the positive cases (probably harmless in the negative/reject but I'd still rather be rid of them)

Downstream programs

These are currently on my GitHub repo philip-schrodt/6HEC/ but can be moved to our eventual public repo.

get_context_models.py

Estimates a set of SVM models from an existing training set—currently designated using the Mkx (e.g. Mk3) infix—for a single context and saves 4 models with the highest F1. Also produces a time-stamped summary file

assess_context.py

Goes through first part (constant `CNTXSAMPLE`) of a set of Release files and writes to a fixed-name .txt file (manually change this name to avoid overwriting when you are satisfied with output) the cases which are all classified by a set of SVM models above a `THRESHOLD` (currently set around 75%ile)

extract_training.py/extract-training-by-field.py:

More specialized: produces a balanced set of context and randomly selected reject cases from output of `plovigy-NGEC-context.py` `extract-training-by-field.py` does the same for the event training files.

assess_outliers

Similar to `assess_context.py` but working with the event training files.

filter-outliers

Looks for consensus in the crime SVMs, gets rid of the ELink HTML strings, and also discards on `len(rec["text"]) <= MIN_LENGTH`, then writes to *Mk8*.

merge_training_edits.py

Fairly simple program with complex I/O that does the initial edit/merge of the output files from `plovigy-NGEC-review.py`, with the output file *"Training-recodes_<category>.json"* containing records for additional processing. It has various JSON records; see internal documentation for details.

split_training_edits.py

splits out the various "plevent" categories—this includes crime, disaster and accident in addition to the PLOVER categories—from this output

combine_annot_stats_b2.py

Generates summary tables from two sets of training cases by combining the output files from ***Colaboratory/*** `estimate_NGEC_evaluate_b2.ipynb`, which is a minor variant of `estimate_NGEC_ensemble.ipynb` which doesn't save the models.

Provenance

Programmer: Philip A. Schrodt
Parus Analytics LLC
Charlottesville, VA, 22901 U.S.A.
http://eventdata.parusanalytics.com