SubString v 1.1

The SubString package consists of three separate command-line modules (substring-A, substring-B and an auxiliary scripts module). Substring-A and -B each implement a different algorithm for consolidating the frequencies of word n-grams of different lengths (i.e. of different *n*). The process of frequency consolidation reduces the frequencies of substrings by the frequencies of the superstrings in which they are contained and an output list is produced showing the consolidated frequencies of all n-grams (see below for more on frequency consolidation). The auxiliary scripts module provides a number of additional functions related to the filtering of n-gram lists. The functions performed by this package will primarily be of interest to linguists and computational linguists working on formulaic language, multi-word expressions and other phraseological phenomena.

A. Overview of modules

substring-A (algorithm with indexation)

This is a python script designed to work in conjunction with the mwetoolkit. It takes as input a corpus-indexed list of n-grams (as produced by mwetoolkit) and then uses an exact, indexation-based algorithm to consolidate frequencies of overlapping n-grams, following Altenberg and Eeg-Olofsson (1990: 16-17).

The Substring-A algorithm is generally preferable to substring-B in cases where access is available to the source texts/corpora from which n-grams are to be extracted, and mwetoolkit can be used to extract n-grams from those texts.

substring-B (algorithm without indexation)

This is a set of Unix shell scripts designed to work on a range of simpler n-gram list formats (where n-grams are not indexed to their occurrences in a specific corpus). Such lists are produced by the NGramProcessor, the Ngram Statistics Package and various other tools and sources such as Google Books Ngrams. The algorithm implemented by substring-B is described in detail in Buerki (2017).

If access to source texts/corpora is unavailable (only n-gram lists are available) or if the limitations of mwetoolkit mean that a different tool needs to be used for n-gram extraction, substring-B should be used. Access to source corpora may be unavailable in case of sources like Google Books Ngrams, or because online corpus portals such as the Sketch Engine are used that allow the creation of n-gram lists, but not the download of the full underlying corpus data. N-gram lists might require some formatting before processing, see the README file for substring-B.

auxiliary scripts

Several auxiliary scripts are included in the SubString package which allow the further processing of n-gram lists after frequency consolidation. These scripts can be used after processing with substring-A or substring-B. See the TUTORIAL included in SubString for details on these scripts.

B. Frequency Consolidation

To illustrate frequency consolidation among n-grams of various lengths, let us assume we have as input the n-grams in (1)a. These will have been extracted from a corpus and their frequency of occurrence in the corpus is indicated by the number to the right of each n-gram.

The 4-gram 'have a lovely time' occurs with a frequency of 15. The 3-grams 'have a lovely' and 'a lovely time' occur 58 and 44 times respectively. 15 of those occurrences are, however, occurrences as part of the superstring 'have a lovely time'. To get the consolidated frequency of occurrence for 'have a lovely' and 'a lovely time' (i.e. the occurrences of these 3-grams on their own, NOT counting when they occur in a longer string), we therefore deduct the frequency of their superstring (15) from their own frequency. This results a consolidated frequency of 43 for 'have a lovely' (i.e. 58 minus 15) and 29 for 'a lovely time' (i.e. 44 minus 15), as shown in (1)b.

The remaining 2-grams ('have a', 'a lovely' and 'lovely time') are also substrings of 'have a lovely time' and therefore also need to have their frequency reduced by 15 (resulting in a frequency of 34692 for 'have a', 86 for 'a lovely' and 30 for 'lovely time'. In addition, 'have a' and 'a lovely' are substrings of 'have a lovely' and therefore the frequency of 'have a lovely' which is now 43, needs to be deducted from their frequencies. This results in a new frequency of 34649 for 'have a' and 43 for 'a lovely'. 'a lovely' and 'lovely time' are furthermore substrings of 'a lovely time' and consequently need to have their frequencies reduced by that of 'a lovely time' (i.e. by 29): the consolidated frequency of 'a lovely' is now 14, that of 'lovely time' is 1. The output of the frequency consolidation is shown in (1)b.

```
(1)a                          (1)b

have a lovely time  15        have a lovely time   15
have a lovely       58        have a lovely        43
a lovely time       44        a lovely time        29
have a           34707        have a            34649
a lovely           101        a lovely             14
lovely time         45        lovely time           1
```

Substring-A implements an algorithm that only consolidates frequencies of n-grams that actually overlap in the source corpus. **Substring-B** has no access to this information and some assumptions are therefore made regarding what sequences are substrings of what other sequences without the possibility of checking this on the basis of indexation to a source text. While consolidations are coherent and accurate with regard to input lists in both algorithms, results of substring-A will inherently differ from those obtained from substring-B.

For more on frequency consolidation among n-grams see Altenberg and Eeg-Olofsson (1990: 16-17), Buerki (2017) and O'Donnell (2011).

C. Dependencies

SubString should run on all systems that can run the bash shell. Substring-A additionally requires Python 3 and an installation of the mwetoolkit 3. For efficient processing of larger amounts of data, substring-B requires bash 4 (although the software will automatically substitute a slower algorithm if only bash 3 is available).[1]

D. Installation

1) Using the supplied installers:

Double-clickable installers are provided for OS X, Linux (with Gnome desktop) and Cygwin/Windows, and an installer script for all UNIX/Linux-like systems. These installers replace previous versions of the SubString package. It may be necessary to log out and log in again before the installation takes full effect.

Gnome Linux / Mac OSX

Inside the SubString directory, double-click on `Linux_installer` (for Ubuntu and other flavours with the Gnome desktop environment) or `OSX_installer` (OS X).

Any Unix/Linux-like system

Open a terminal window, drop the `install.sh` script (located inside the `bin` directory) onto the terminal window and press ENTER. This will start the interactive installation process.

Windows

The Cygwin environment needs to be installed first. During the installation procedure for Cygwin, the optional 'bc' package from the 'maths' category will need to be installed as well as the 'python3' package from the 'Python' category. A guide on how to install Cygwin is found here.

After Cygwin has been installed, double click on the `Cygwin_installer` or `Cygwin64_installer` (try both if one does not work) to start the installation process.

Especially under Windows systems, there can be problems if file and/or folder names contain spaces, so it is best to avoid locations on the system that have folder names with spaces.

2) Manual installation / other flavours of Linux:

To install the substring-A module, copy `substring-A.py` (inside the `bin` directory) to the same bin directory as the other mwetoolkit files; `ft_txtcandidates.py` should be placed in the `bin/mwetk/filetype` directory of the mwetoolkit installation. The location should be in the users $PATH variable.

All other scripts in the `bin` folder should be placed in a location that is in the user's $PATH variable (or the location should be added to the $PATH variable) so they can be called from the command line. A good place to put the scripts might be `/usr/local/bin` or `~/bin` .

To uninstall, open a terminal window then drop the `install.sh` script (located inside the `bin` directory) on to the terminal window, then type `-u` so that the line ends in `install.sh -u` and press ENTER. Alternatively, manually delete the relevant files from their installation locations as indicated above.

E. Basic Operation

The frequency consolidation modules A and B can be accessed in interactive mode by double-clicking on the SubString icon (If an icon was generated on the desktop or in the applications folder during installation). Alternatively, it can be started from a terminal by typing `substring.sh` and pressing ENTER. In non-interactive mode, the three modules of SubString are used as shown below. See the TUTORIAL for examples and more information.

substring-A module

This module is invoked from the command line like so `substring-A.py [OPTIONS] INPUT_FILE.xml` ; options are displayed by running `substring-A.py -h` and input files must be in mwetoolkit's XML format.

substring-B module

Substring-B is probably best used interactively, but can be invoked from the command line like so

`substring-B.sh [OPTIONS] [-u uncut_list]+ INPUT_FILE.txt+`, where `INPUT_FILE.txt+` is two or more n-gram lists of different length (use `-h` or the TUTORIAL for usage details).

N-gram lists processed by the B module must be formatted in the correct format (see `test_data-B` folder for examples). N-gram lists produced by the N-Gram Processor are suitable for direct input; if n-gram lists were produced by a different programme they might need reformatting.

auxiliary scripts

- `cutoff.sh -f N FILE+` to apply a frequency cutoff to an n-gram list
- `length-adjust.sh -c N [-OPTIONS] FILE+` see TUTORIAL for details
- `en-filter.sh [-OPTIONS] FILE+` to apply a structural filter to n-gram lists
- `random_lines.sh [-OPTIONS] FILE` to extract random lines from a list
- `TP-filter.sh [-OPTIONS] FILE` to assess the precision of a filtered list of n-grams
- `listconv.sh [-OPTIONS] FILE[S]` to convert n-gram lists between different formats

These scripts accept n-gram lists in the format of the N-Gram Processor which is also the format of substring-B and can be specified as output format to substring-A by specifying the option `--to NGP`.

See the TUTORIAL file included in the SubString directory for examples and more information on all three modules.

F. Known Issues

On OSX it's been reported that on some systems, clicking on the SubString icon in the applications folder or desktop brings up the following error: 'SubString can’t be opened because Sandbox is not allowed to open documents in Terminal.' If this happens, delete the SubString app that brings up this message and instead use the file `SubString` (with the icon of a black terminal window in a white frame) to launch SubString interactively. This file is located in the `bin` directory within the SubString directory.

Issues can be raised at http://github.com/buerki/SubString/issues, but no active support can be provided for this software.

G. Copyright, licensing, download

The project resides at http://buerki.github.com/SubString/ and new versions will be posted there. To be notified of new releases, go to https://github.com/buerki/SubString, click on the 'Watch' button and sign in.

H. Warning

As article 7 of the EUPL states, the SubString is a work in progress, which is continuously improved. It is not a finished work and may therefore contain defects or “bugs” inherent to this type of software development. For the above reason, the software is provided under the Licence on an “as is” basis and without warranties of any kind concerning it, including without limitation merchantability, fitness for a particular purpose, absence of defects or errors, accuracy, non-infringement of intellectual property rights other than copyright as stated in Article 6 of this Licence. This disclaimer of warranty is an essential part of the Licence and a condition for the grant of any rights to SubString.

---

1. Most recent operating system versions have bash v. 4 installed as standard, but MacOS X has bash v. 3.2 installed as standard. Bash v. 4 can be installed using MacPorts, Homebrew or similar and then the new version would either need to be put in the directory `/bin` (replacing the old version) or the first line of the `substring-B.sh` script would need adjusting to point to the new version of bash (if installed via MacPorts, the new line would read `#!/opt/local/bin/bash` instead of `#!/usr/bin/env bash`). ↩