Tool for the Automatic Analysis of Syntactic Sophistication and Complexity (TAASSC)

User Manual for TAASSC 1.1 (updated 4-26-2017)

Kristopher Kyle - University of Hawaii at Manoa
Scott Crossley - Georgia State University

This document is intended to assist users of TAASSC. It includes a brief explanation of how to use the tool. Additional information about TAASSC is included in the supplementary Index Description Spreadsheet (available at www.kristopherkyle.com).

Please use the following citation when referencing TAASSC in your work:

Kyle, K. (in press). *Measuring syntactic development in L2 writing: Fine grained indices of syntactic complexity and usage-based indices of syntactic sophistication*. (Doctoral dissertation).

If you use any of the SCA indices, please also use the following citation:

Lu, X. (2010). Automatic analysis of syntactic complexity in second language writing. *International Journal of Corpus Linguistics*, 15(4):474-496.

Getting Started

TAASSC is freely available at http://www.kristopherkyle.com/taassc.html. Download the version that is appropriate for your operating system.

For TAASSC to work properly, you must also download and install the Java Development Kit (JDK), which is freely available at: http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html.

Options

TAASSC analyzes a number of textual aspects related to syntactic complexity and sophistication. The user can choose to include indices related to all of these aspects, or can choose to constrain the output.

The user may choose to include indices from any (or all) of the following categories:

- clause complexity
- phrase complexity
- syntactic sophistication
- aggregated component scores related to clause complexity, phrase complexity, and syntactic sophistication
- classic indices of syntactic complexity as measured by the L2 Syntactic Complexity Analyzer (L2SCA) version 3.3.3

    - Note that including SCA indices will significantly increase processing time. Consider running other desired indices first, then running SCA indices separately.

The user may also choose to output files processed by TAASSC and SCA for follow-up analyses (advanced). Output may include:

- text file databases, which include information pertaining to each clause and/or phrase in your target texts in tab-delimited format
- parsed versions of each of your target texts in xml format
    - TAASSC indices use "collapsed-ccprocessed-dependencies" in the xml files included in the "mod_parsed" folder
    - SCA indices use the parse trees in the xml files included in the "sca_parsed" folder

Input

All input files must be text files (.txt) that do not include any type of markup (e.g., XML, HTML, etc.). Files must be located in a single folder. TAASSC will process all .txt files in the chosen input folder. Please note the following when formatting your files:

- Input text filenames should not include spaces, quotation marks, or commas
- For best results, your texts should be in sentence case (i.e., not all lower-case or upper case).
- Any and all text in a file will be processed and factored into index scores. For accurate results, make sure no unwanted headers are included in the input files.

Saving Your Output

TAASSC provides output in the form of a comma-separated (.csv) file that can be opened with any spreadsheet software. The default output file name is "results.csv", though we would recommend changing this file name each time you run TAASSC to ensure that the file is not overwritten. Syntactic components and SCA indices will be included in separate .csv files (i.e., "results_components.csv" and "results_sca.csv").

Indices

TAASSC 1.0 calculates 372 indices in five categories: Clause complexity (32 indices), phrase complexity (132 indices), syntactic sophistication (190 indices), syntactic component scores (9 indices), and classic syntactic complexity indices (14 indices). Please see the supplementary spreadsheet file for more information. Also, see the dissertation referenced above for more information.

Use of component scores

Note that the component scores are computed based on standardized scores. This means that if one plans to compare data with regard to these scores, ALL data included in the analysis must be processed by TAASSC at the same time. For example, if one were to compare high and low quality essays, both the high and the low quality essays would need to be placed in the same folder to be processed by TAASSC.

Visualization (Advanced)

Also included at www.kristopherkyle.com is a visualization tool for analyzing dependency relations in TAASSC processed xml files (i.e., files included in the "mod_parsed" folder). This visualization is implemented using brat and requires some programming knowledge to use. To try it out, download the visualization folder from www.kristopherkyle.com and follow the instructions in the readme file.

SCA output can be visualized using Tregex. One can enter the Tregex search patterns used by SCA (e.g., clause, complex nominals, etc.), which are found in the Python script for SCA, to see what is identified.