Computerized Linguistic Analysis System (CLAS)

The Computerized Linguistic Analysis System (CLAS; http://rxinformatics.umn.edu/clas.html) is designed to evaluate written language output for syntactic complexity. So far, three approaches to measuring complexity have been implemented: Yngve, Frazier and Syntactic Dependency Length. The current implementation relies on the output of the Stanford PCFG Parser (2011-06-08 release) (http://nlp.stanford.edu/software/lex-parser.shtml). The parser uses a PCFG grammar trained on a Wall Street Journal corpus (wsjPCFG.gz).

The CLAS tool may encounter errors in processing files with a blank line at their end. If this is your case, please, open the file with a text editor and delete the last blank line.

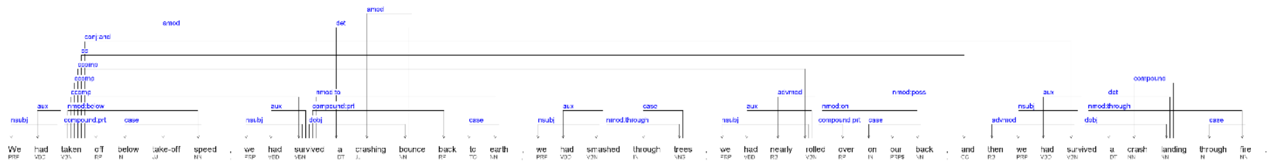
Output paths with spaces in them may also cause the CLAS tool to fail exporting the output. Please, always make sure that the selected output folder path does not contain spaces.

To compare syntactic complexity scores across different sets of documents (e.g., different books), Pakhomov et al. recommend using one-way ANOVA with subsequent pairwise post-hoc t-tests using Tukey's Honestly Significant Differences (HSD) approach to adjust for multiple comparisons.

fdepth (Frazier) and ydepth (Yngve) represent something very different from syntactic dependency length and it may not be surprising if they point in opposite directions.

In the Excel output fie, the denominator for the mean scores is not provided by CLAS. The mean for example for the Fdepth is calculated by summing up the depths of each token in the parse tree (totalFdepth) and dividing by the number of tokens. The number of tokens, unfortunately is also not given by the CLAS output. It is basically the number of words in each sentence plus all the punctuation marks. For example, the total_Fdepth for the sentence "Miracles thicker than fog on flight of No 10607." is 11.5: the sentence has 10 tokens (8 words, one number (10607) and one period. So, 11.5/10 = 1.15 - mean_Fdepth. For the sentence "We had taken off below take-off speed, we had survived a crashing bounce back to earth, we had smashed through trees, we had nearly rolled over on our back, and then we had survived a crash landing through fire.", the total_Fdepth is 44 and the sentence has 46 tokens (the word "take-off" is split into 3 tokens by the parser "take" "-" "off"). 44/46 = 0.9565 - mean_Fdepth, and so on.

Syntactic dependency mean is calculated yet differently - the sum of all dependency lengths is divided by the total number of dependencies (again, that denominator is not printed out to the CLAS output). If you trust CLAS calculations, you can get the number of dependencies by simply dividing total_SynDepLen by mean_SynDepLen. For sentence "We had taken off below take-off speed, we had survived a crashing bounce back to earth, we had smashed through trees, we had nearly rolled over on our back, and then we had survived a crash landing through fire.", this would be 158/4.9375 = 32 dependencies. CLAS uses a version of the Stanford CoreNLP parser, so the number of dependencies may not match with, for instance, a png graph of dependency tree.



Note that some of the dependencies in this graph are super long - there is one that's about 32 tokens long (number of tokens found in between). This may lead to a large complexity number on this sentence. Because the independent clauses are separated with commas, the parser "thinks" that for example the conjunction "and" in "and then we had survived" joins "taken" and "survived". The way this sentence is put together, the independent clauses separated by commas are treated essentially as "concepts" - might as well be a bunch of noun phrases. In that sense, this is indeed a VERY complex sentence to process but it masquerades in a syntactic structure that appears to be deceptively simple but is really not!

A more detailed description of the CLAS system along with results of an experiment involving the analysis of Isis Murdoch writings can be found in this publication:

Pakhomov, S., Chacon, D., Wicklund, M., Gundel, J. (2010). Computerized Assessment of Syntactic Complexity in Alzheimer's Disease: A Case Study of Iris Murdoch's Writing. Behavioral Research Methods. 43(1):136-144.

Additional information on these measures and their use in dementia research can be found here:

Roark, B., Mitchell,M., & Hollingshead, K. (2007). Syntactic complexity measures for detecting Mild Cognitive Impairment. Paper presented at the ACL 2007 Workshop on Biomedical Natural Language Processing (BioNLP), Prague, Czech Republic.

CLAS has been recently (June 19, 2011) refactored. The system still uses the Apache UIMA platform; however, the refactored code now relies on three more open source projects that make it easier to work with UIMA:

1. uimaFit - http://code.google.com/p/uimafit/
2. clearTk - http://code.google.com/p/cleartk/
3. Biomedicus - http://code.google.com/p/biomedicus/

Make sure that sentences are marked with a period followed by a space or CLAS will fail. CLAS uses a probabilistic sentence detection algorithm which tends to be fairly accurate if the sentence boundaries are more or less clear. The output directory (./example/output/) will be populated with two output files for each input file - .xmi and .dat file. The xmi file contains all annotations (e.g., tokens, sentence boundaries, POS, etc.) in a serialized form. These xmi files are readable with tools that come with the Unstructured Information Management Architecture (UIMA). In particular, the UIMA CAS Visual Debugger can be used to read in the xmi files and visualize annotations and features.

The .dat files written to the output directory are pipe-delimited tables that contain the values of the measurements made during parsing. The first row contains variables names. Here is the format and the meaning of the labels:

```
sentenceID|sentence_text|num_clauses|mean_Fdepth|total_Fdepth|mean_Ydepth|total_Ydepth|mean_SynDepLen|total_SynDepLen|noun_count|adj_count|adverb_count|verb_count|det_count|conj_count|prep_count|properN_count
```

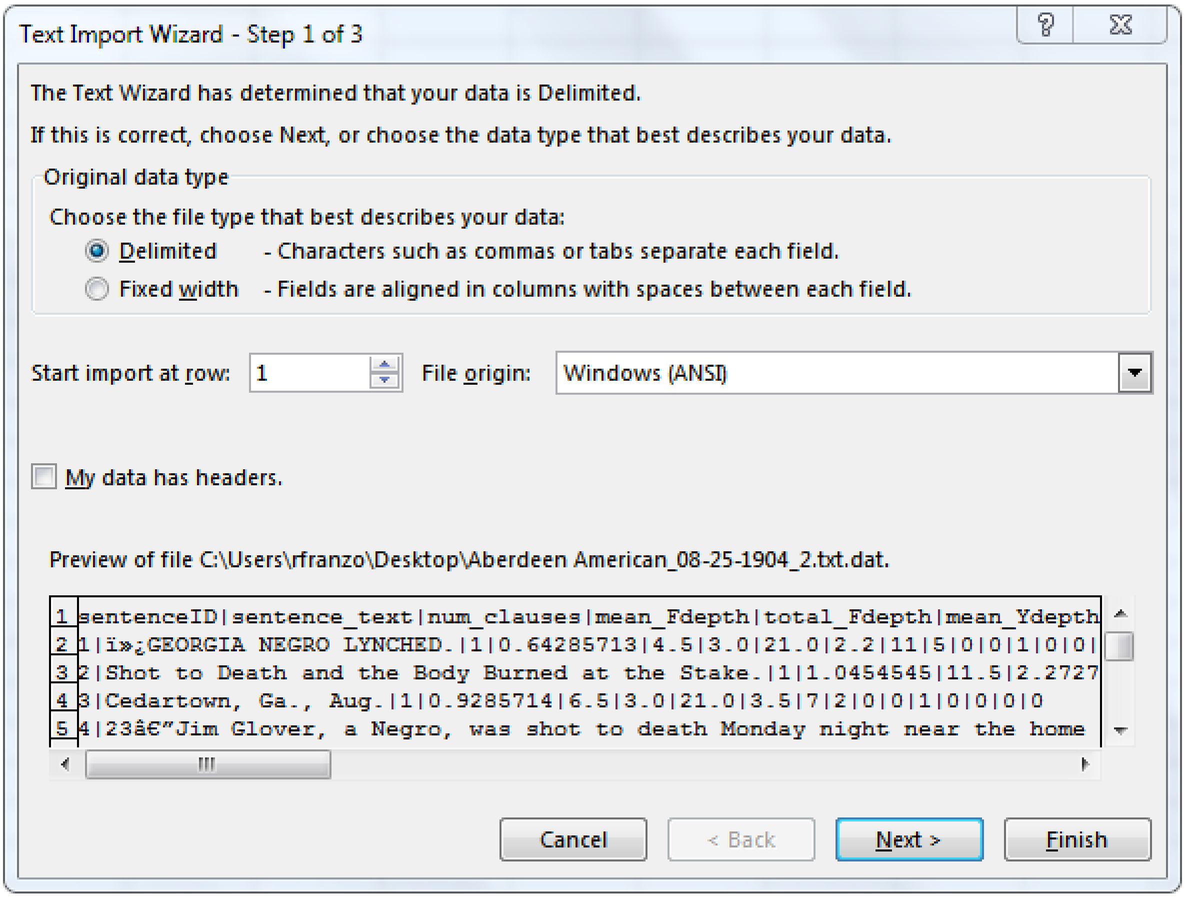
sentenceID - numerical id (integer) of the sentence in the order of appearance
sentence_text - text of the sentence
num_clauses - number of "S" nodes in the parse tree
mean_Fdepth - mean of Frazer depth scores on individual tokens
total_Fdepth - sum of Frazer depth scores on individual tokens
mean_Ydepth - mean of Yngve depth scores on individual tokens
total_Ydepth - sum of Yngve depth scores on individual tokens
mean_SynDepLen - mean of syntactic dependency lengths in the dependency parse
total_SynDepLen - sum of syntactic dependency lengths in the dependency parse
noun_count - raw count of nouns
adj_count - raw count of adjectives
adverb_count - raw count of adverbs
verb_count - raw count of verbs
det_count - raw count of determiners
conj_count - raw count of conjunctions
prep_count - raw count of prepositions
properN_count - raw count of proper nouns

The output files can be read into R statistical software package using the following command executed from an R console window:

```
data <- read.table("./example/output/001.txt.dat",header=TRUE,sep="|")
```

To view the results you should import the “.dat” file in excel selecting File, Open and following the steps below.

First choose to import the data from delimited file. Then select the delimiter used, the pipe character in this case “|”.



Text Import Wizard - Step 1 of 3

The Text Wizard has determined that your data is Delimited.

If this is correct, choose Next, or choose the data type that best describes your data.

Original data type

Choose the file type that best describes your data:

● Delimited - Characters such as commas or tabs separate each field.

○ Fixed width - Fields are aligned in columns with spaces between each field.

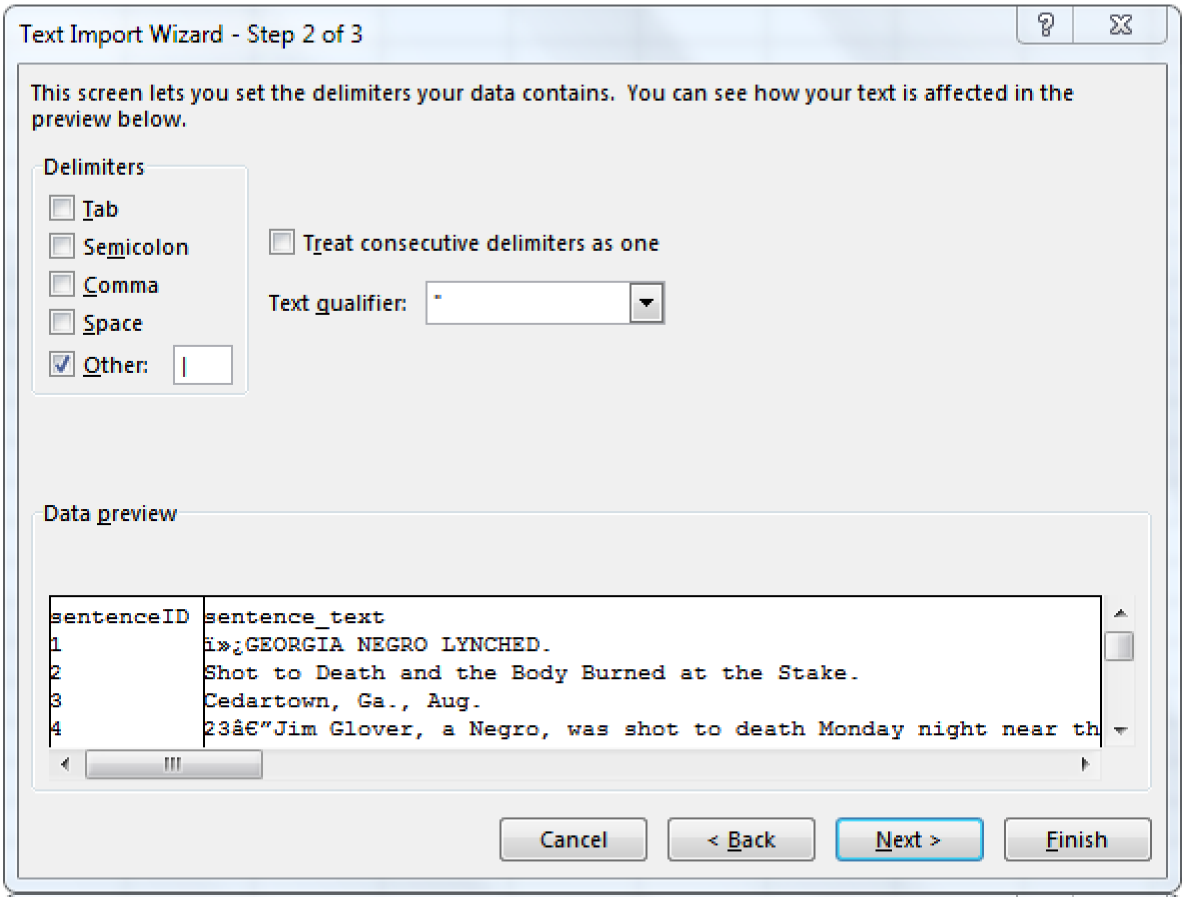
Start import at row: 1

File origin: Windows (ANSI)

□ My data has headers.

Preview of file C:\Users\rfranzo\Desktop\Aberdeen American_08-25-1904_2.txt.dat.

```
1 sentenceID|sentence_text|num_clauses|mean_Fdepth|total_Fdepth|mean_Ydepth
2 1|ï»¿GEORGIA NEGRO LYNCHED.|1|0.64285713|4.5|3.0|21.0|2.2|11|5|0|0|1|0|0|
3 2|Shot to Death and the Body Burned at the Stake.|1|1.0454545|11.5|2.2727
4 3|Cedartown, Ga., Aug.|1|0.9285714|6.5|3.0|21.0|3.5|7|2|0|0|1|0|0|0|0
5 4|23â€”Jim Glover, a Negro, was shot to death Monday night near the home
```



Text Import Wizard - Step 2 of 3

This screen lets you set the delimiters your data contains. You can see how your text is affected in the preview below.

Delimiters

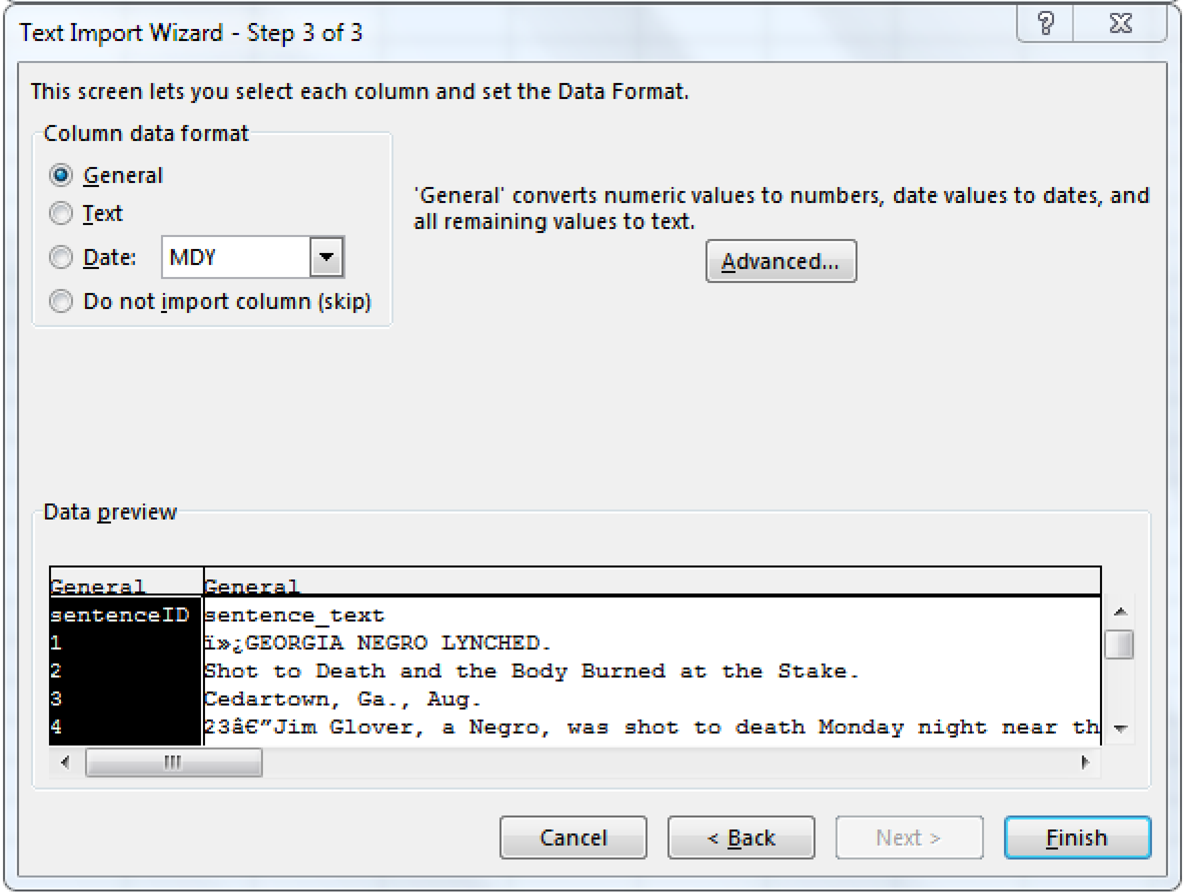
- ☐ Tab
- ☐ Semicolon
- ☐ Comma
- ☐ Space
- ☑ Other: |

☐ Treat consecutive delimiters as one

Text qualifier: "

Data preview

```
sentenceID sentence_text
1          ï»¿GEORGIA NEGRO LYNCHED.
2          Shot to Death and the Body Burned at the Stake.
3          Cedartown, Ga., Aug.
4          23â€”Jim Glover, a Negro, was shot to death Monday night near th
```



Text Import Wizard - Step 3 of 3

This screen lets you select each column and set the Data Format.

Column data format

- ● General
- ○ Text
- ○ Date: MDY
- ○ Do not import column (skip)

'General' converts numeric values to numbers, date values to dates, and all remaining values to text.

Data preview

```
General    General
sentenceID sentence_text
1          ï»¿GEORGIA NEGRO LYNCHED.
2          Shot to Death and the Body Burned at the Stake.
3          Cedartown, Ga., Aug.
4          23â€”Jim Glover, a Negro, was shot to death Monday night near th
```