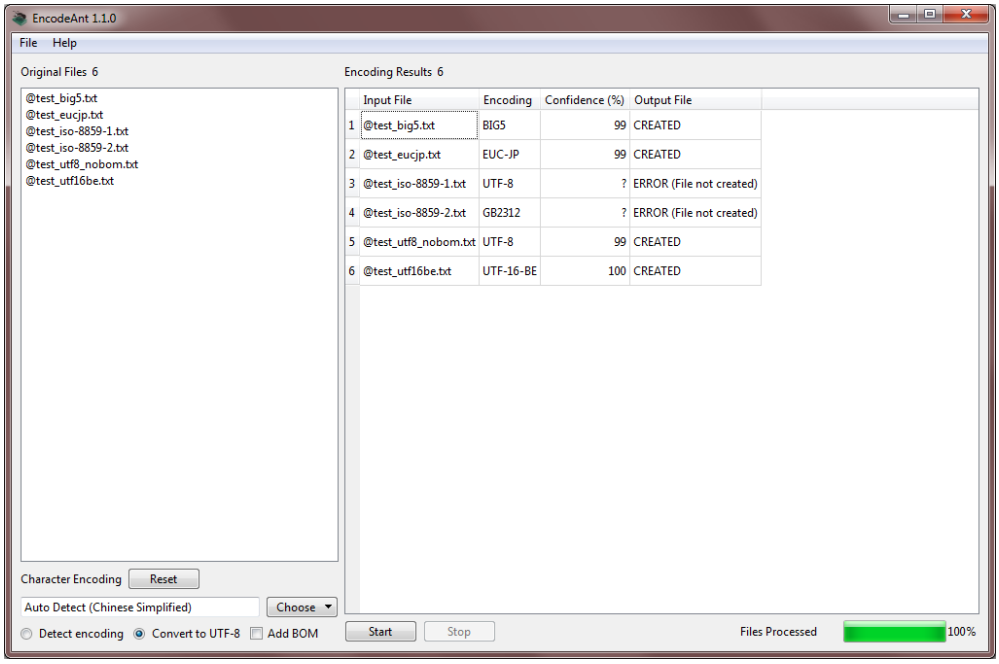
EncodeAnt (Windows)

Build 1.2.0

Laurence Anthony, Ph.D.
Center for English Language Education in Science and Engineering, School of Science and Engineering, Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan.
Help file version: 001 (May 14, 2016).



Introduction

EncodeAnt is a freeware character encoding detection and conversion tool. *EncodeAnt* takes an input list of text files (e.g. .txt) and attempts to auto-detect the character encoding that the files use. The character encoding can also be set manually. *EncodeAnt* also has an option to auto-convert the character encoding of the files to UTF-8, which is a standard used in most corpus research. The converted files are saved in a separate folder leaving the original files untouched.

EncodeAnt runs on any computer running Microsoft Windows (tested on Win 98/Me/2000/NT, XP, Vista, Win 7, Win 8). It is developed in Python and Qt using the *PyInstaller* compiler to generate executables for the different operating systems.

Getting Started (No installation necessary)

Windows

On Windows systems, simply double click the *EncodeAnt* icon to launch the program.

Step 1: Select the files you want to analyze. You can do this in three ways:

a) Click on the File->Open File(s) menu option and select the files you want to segment;

b) Click on the File->Open Dir menu option and select a directory of files you want to segment;

c) Drag and drop files directly onto the *EncodeAnt* application.

Note 1: The number of selected files is shown next to the "Original Files" label.

Note 2: If you click on the File->Close Files menu option, the files will removed from the list.

Step 2: Choose one of the character encoding options. If you choose one of the "auto detect" options, *EncodeAnt* will attempt to guess the encodings of the files. If you choose one of the "common" or "all" options, AntConc will attempt to use this encoding choice.

Step 3: Choose the "Detect encoding" radiobox to detect the encodings of your files or the "Convert to UTF-8" radiobox to convert the encodings of your files (based on the character encoding option you chose in step 2) to the standard UTF-8 encoding.

Step 4: Click the "Add BOM" checkbox if you want to add a UTF-8 BOM (Byte Order Mark) to the beginning of your converted files from step 3. This is generally not recommended but it is useful on Microsoft Windows systems that use the BOM to open the file correctly in programs like Notepad.

Step 5: Click "Start" to begin the detection/conversion process.

Additional Features

The output display can be selected, copied, and pasted as is standard on the operating system:

Windows: CTRL-A ⇨ Select All CTRL-C ⇨ Copy CTRL-V ⇨ Paste

NOTES

General points

- If a "common" or "all" encoding option is chosen, the results will report a confidence of the encoding as 100% (even in cases when the actual encoding of the file is different). If an "auto-detect" option is chosen, the confidence will be reported on a scale of 0% to 100% or given a value of "?" when the confidence cannot be determined. (See the notes on detection methods below).
- If the "Convert to UTF-8" option is selected, the converted files will be saved where the original file was stored under a sub-folder called "utf8".
- The detection/conversion process can be stopped at any time by clicking the "Stop" button.

Detection methods used by *EncodeAnt*

- The first method attempts to read BOMs at the beginning of files. These BOM unambiguously decide the encoding.
- The second method uses Chardet (https://pypi.python.org/pypi/chardet) to guess the encoding. This tool produces confidence measures for the guess, which are reported in the results table.
- The third method uses Chared (https://code.google.com/p/chared/). This relies on knowing the target (human) language of text in the file. If the first method is unsuccessful or the confidence level of the second method is too low, this third approach is adopted.

COMMENTS/SUGGESTIONS/BUG FIXES

All new editions and bug fixes are listed in the revision history below. However, if you find a bug in the program, or have any suggestions for improving the program, please let me know and I will try to address the issues in a future version.

This software is available as 'freeware' (see Legal Matter below), but it is important for my funding to hear about any successes that people have with the software. Therefore, if you find the software useful, please send me an e-mail briefly describing how it is being used.

CITING/REFERENCING *EncodeAnt*

Use the following method to cite/reference *EncodeAnt* according to the APA style guide:

Anthony, L. (YEAR OF RELEASE). *EncodeAnt* (Version VERSION NUMBER) [Computer Software]. Tokyo, Japan: Waseda University. Available from http://www.laurenceanthony.net/

For example if you download *EncodeAnt 1.1.0*, which was released in 2014, you would cite/reference it as follows:
Anthony, L. (2014). *EncodeAnt* (Version 1.1.0) [Computer Software]. Tokyo, Japan: Waseda University. Available from http:// www.laurenceanthony.net/

Note that the APA instructions are not entirely clear about citing software, and it is debatable whether or not the "Available from ..." statement is needed. See here for more details:
http://owl.english.purdue.edu/owl/resource/560/10/

LEGAL MATTER

EncodeAnt can be used freely for individual use for non-profit research purposes, and freely distributed on the condition that this read me file is attached in an unaltered state. If the software is planned to be used in a group environment, you are required to inform me how the software is to be used, and I will then determine if you can have permission to use it. The software comes on an 'as is' basis, and the author will accept no liability for any damage that may result from using the software.

KNOWN ISSUES

None

REVISION HISTORY

1.2.0

A minor upgrade improving performance and sensible handling of malformed characters in the input files.

New features

1. The program runs much quicker by checking only the first 300 characters of a file instead of reading the entire file into memory.
2. The program now replaces malformed characters in both the input and output files with standard replacement characters.

Bug fixes

1. Dragging and dropping a file on to the program now correctly updates the input file counter.
2. Dropping a folder (instead of files) on to the program now correctly ignores the drop.

1.1.0

A minor upgrade featuring new features.

New features

3. The program can now convert files based on a manually selected character encoding or one of three auto-detection approaches.
4. The program now includes three different auto-detect systems (see the notes above)
5. The program now reports when files could not be converted correctly either because they contained no text data, the manually set character encoding produced an error when applied, or the auto-detected character encoding produced an error when applied.
6. Files loaded into the program can now be closed individually (by selection) or all at once.
7. The file selection window now has more text selection options.
8. Encodings are now always displayed in ALL-CAPS.

1.0.0
This is the first version of the program