

Literary Toolbox IV

Name: Literary Toolbox IV
Author: Mike Pasini
Version: 1.0a
Last Update: 17 August 2022
Requires: Keyboard Maestro v10
Optional: Various Perl modules, Nisus Thesaurus, Spell Catcher, Internet access

Table of Contents

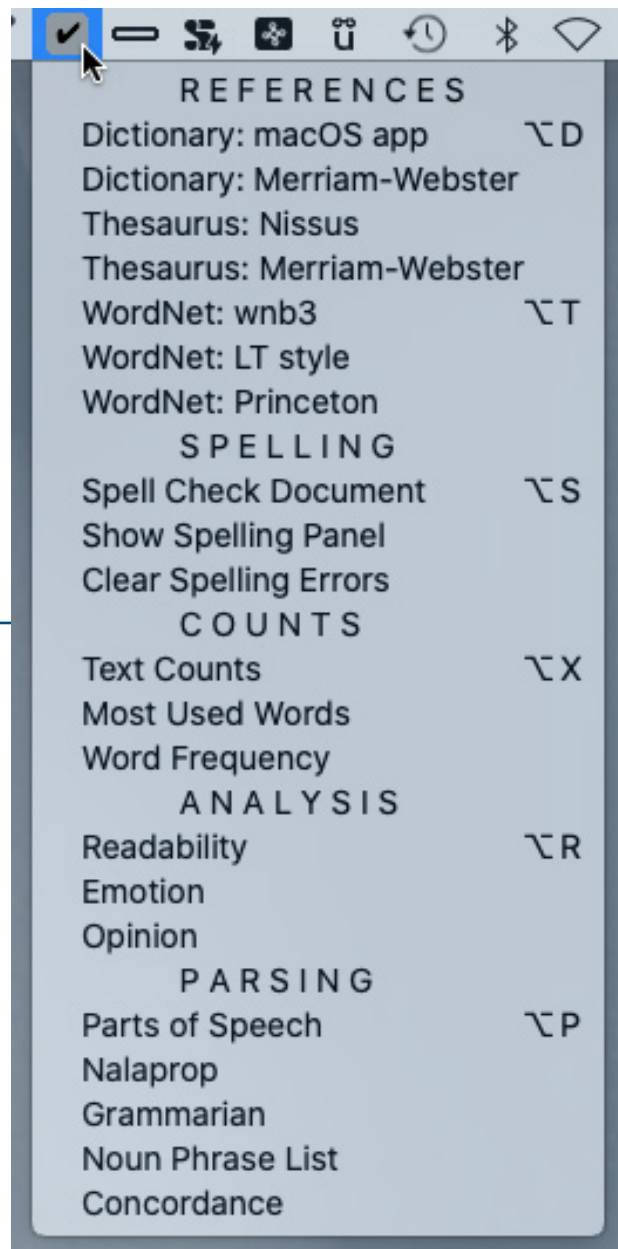
Intro
Description
What's New
Installation
Uninstalling Modules
A Palette Theme
Notes
Release Notes
Support

INTRO

This set of macros can be used to evaluate text in several different ways. They can be helpful to authors but also to readers evaluating or studying various texts.

Since Literary Toolbox III, all supporting code is 64-bit. Literary Toolbox has been developed and tested on macOS Catalina 10.15.7.

With the introduction of a subroutine, this version requires Keyboard Maestro v10 but instructions are provided in the code can be modified easily enough to run under the previous version of Keyboard Maestro.



Each option works on a text selection rather than a discrete file, parsing the selection copied to the clipboard and stripping it of any style information before analysis. That helps (if isn't quite a cure) for text selected in a word processor. But you should have no problem with text selections made in any editor or within a Web browser.

If a selection has not been made, Select All is employed to create one.

Some of the tools push the limits of what an algorithm can tell us about human communication but we still find them useful if not for raw data at least for comparisons or trends.

In fact, Literary Toolbox has always considered redundancy a hedge against inaccuracy. Algorithms to count words, to take one example, are inherently imprecise but when you use two or three of them you get a more accurate picture.

So this update adds another dictionary and thesaurus making two dictionaries and three thesauri available.

We've used some of these for years and some for a few months. And while they do tap into the power of some third-party code it's free code.

Keyboard Maestro provides a simple interface to these tools. A palette (with an optional custom color scheme) organizes them for easy recall. And custom HTML forms that use the same color scheme as the palette provide easy-to-read reports. Options to use online resources return results in a self-contained browser window.

Literary Tools II changed how the original Literary Tools delivered text to these modules. Originally the text was stored in a Keyboard Maestro variable. But long texts that would have overrun the environment space were ignored, so the newer version wrote texts of any length to a temporary file at /tmp//LT-temp.txt, which is immediately deleted after it is read. Literary Tools III and IV use that same approach.

DESCRIPTION

The zip file contains several macros grouped into five sections on one palette. Many of the macros execute Perl scripts that report via custom HTML windows.

Perl 5, required for the scripts, has traditionally been included in all OS X installations. The modules required to extend Perl 5 for the Literary Toolbox are free.

The palette also taps into a free thesaurus from Nisus and the built-in OS X Dictionary. It uses macOS system spelling utilities when available.

It also includes built-in access to online resources including the Merriam-Webster dictionary and thesaurus as well as WordNet using Princeton University's servers.

The [References](#) section contains:

- **Dictionary: macOS app:** to look up the currently selected word in the macOS Dictionary application (which can include several sources as well as a thesaurus)

included with various versions of the Macintosh operating system. It can optionally (⌘D) look up a selection in the online Merriam-Webster dictionary.

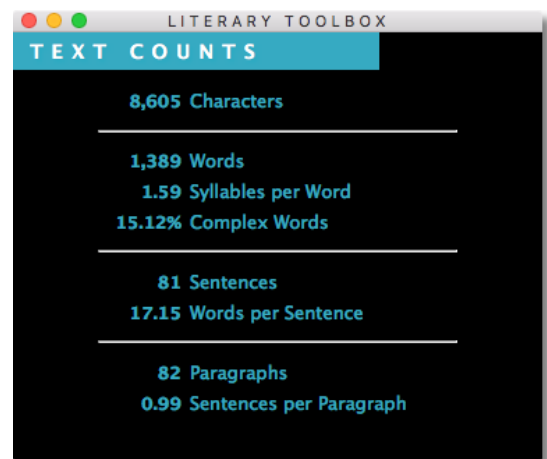
- **Dictionary: Merriam-Webster** : to look up the currently selected word online in the Merriam-Webster dictionary.
- **Thesaurus: Nissus**: to look up the currently selected word in the free Nissus Thesaurus (<https://nissus.com/Thesaurus/>) containing over 120,000 words.
- **Thesaurus: Merriam-Webster**: to look up a selection in the online Merriam-Webster thesaurus.
- **WordNet:wmb3**: to look up a word in a local copy of Princeton's dictionary/thesaurus using the free wmb3 app by Sergey Sikorskiy.
- **WordNet: LT style** : to look up a word in Princeton's online dictionary/thesaurus, returning the results in a window that matches the other modules in the toolbox. There are, however, no links to follow.
- **WordNet: Princeton**: to look up a word in Princeton's online dictionary/thesaurus. Results are displayed in a self-contained browser window so you can continue to explore the database

The **Spelling** section contains:

- **Spell Check Document**: to enable spell checking in the document using the system to underline suspicious words in red.
- **Show Spelling Panel**: to check spelling in the document using the macOS spell checker panel in applications that support it to go word by word through the list of suspicious characters.
- **Clear Spelling Errors**: to remove the system's red underlining of suspicious words, which can be distracting when not spell checking.

The **Counts** section contains:

- **Text Counts**: provides eight counts of characters, words, syllables per word, complex words, sentences, words per sentence, paragraphs and sentences per paragraph for the selected text, helpful in navigating toward a better readability score.
- **Most Used Words**: to display a list of the most frequently used words (anything used more than once) in your selection. The list is sorted by number of occurrences over a single occurrence and omits small words like "the, and, or, an, a, in, to, of, on,



for, by, in, but."

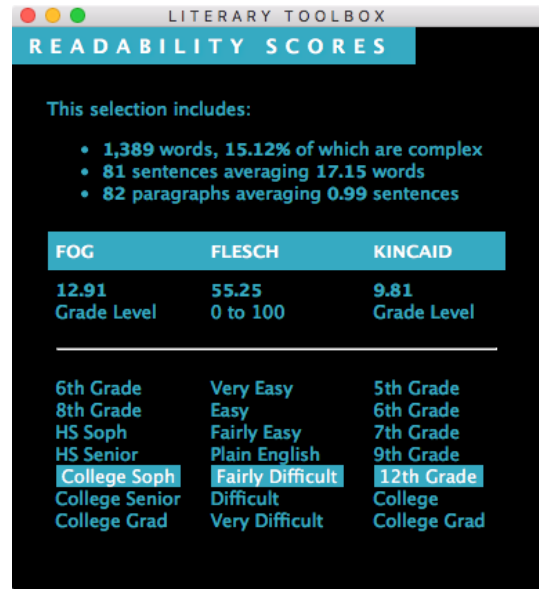
- **Word Frequency:** to display an alphabetical list of all the words in a selection with their count.

The **Analysis** section contains:

- **Readability:** to report readability scores for the Fog, Flesch and Flesch-Kincaid indices both by number and on a scale of reading difficulty or grade level. Some related counts are also reported.
- **Emotion:** to display the sentences that most evince several emotions.
- **Opinion:** to display sentences that are positive or negative.

The **Parsing** section contains:

- **Parts of Speech:** a colorized version of the selection in which each word is colored by its part of speech. Includes counts for each part of speech. Hover over a word in the Tagged Text section to see its precise part of speech, which is a bit more detailed than the general form indicated by the color. When invoked with the



FOG	FLESCHE	KINCAID
12.91	55.25	9.81
Grade Level	0 to 100	Grade Level
6th Grade	Very Easy	5th Grade
8th Grade	Easy	6th Grade
HS Soph	Fairly Easy	7th Grade
HS Senior	Plain English	9th Grade
College Soph	Fairly Difficult	12th Grade
College Senior	Difficult	College
College Grad	Very Difficult	College Grad

LITERARY TOOLBOX

TAGGED TEXT LEGEND

LEGEND		
CC	Conjunction, coordinating	and, or
CD	Adjective, cardinal number	3, fifteen
DET	Determiner	this, each, some
EX	Pronoun, existential there	there
FW	Foreign words	niente
IN	Preposition/Conjunction	for, of, although, that
JJ	Adjective	happy, bad
JJR	Adjective, comparative	happier, worse
JJS	Adjective, superlative	happiest, worst
LS	Symbol, list item	A, A.
MD	Verb, modal	can, could, 'll
NN	Noun	aircraft, data
NNP	Noun, proper	London, Michael
NNPS	Noun, proper, plural	Australians, Methodists
NNS	Noun, plural	women, books
PDT	Determiner, prequalifier	quite, all, half
POS	Possessive	's, '

LITERARY TOOLBOX

TAGGED TEXT

COUNTS

54 Adjectives	99 Nouns	42 Pronouns
15 Adverbs	90 Verbs	51 Prepositions
15 Conjunctions	51 Determiners	39 Punctuation
Possessives	Symbols	Foreign Words
Interjections	Untagged	

TAGGED TEXT

The built-in Dictionary provides a wealth of information often sufficient for my own needs but WordNet is an intriguing alternative. The free Thesaurus option from Nisus is local but both Dictionary and WordNet provide local options in this release. You can access WordNet if you are off-line if you install wnb3.app from <https://www.sikorskiy.net/info/prj/wnb3/>, which includes the data files. You can use the dark mode formatted report for you initial display with the third WordNet option. You can just pick one and stick with it, eliminating the others. I had at first wanted to do a local dark mode option but that, I learned, would require installing node.js, which is n't something I wanted to support. I may still do it because I prefer the dark mode interface to the other two, so if you're interested rattle my chain. The built-in Dictionary provides a wealth of information often sufficient for my own needs but WordNet is an intriguing alternative. The free Thesaurus option from Nisus is local but both Dictionary and WordNet provide local options in this release. You can access WordNet if you are off-line if you install wnb3.app from <https://www.sikorskiy.net/info/prj/wnb3/>, which includes the data files. You can use the dark mode formatted report for you initial display with the third WordNet option. You can just pick one and stick with it, eliminating the others. I had at first

Option key, an extended color-coded legend of these codes appears in a separate window to the left (as soon above).

- **Nalaprop:** to analyze the selection using macOS's text analysis tools introduced in Catalina Howard Oakley's Nalaprop. Literary Toolbox reads the text it copies to a temporary file for analysis by Nalaprop.
- **Grammarian:** to check your selection for grammar errors and cautions with totals for each (see below). If the macro is invoked with the **Option** key, Cautions are skipped. This version of the Toolbox quotes your text to make it easier to find the reference. Note that paragraphs are not always accurately detected so using the Concordance below (especially with Cautions) can be helpful.
- **Noun Phrase List:** to display noun phrases sorted by number of occurrences.
- **Concordance:** display sentences which contain a particular word or phrase. This can show how frequently a word appears in a text as well as help track down a fuller context for the Cautions cited by Grammarian.

Error	Graf	Explanation	Suggestion
has become	5	Incorrect use of infinitive instead of past participle » ?	has become
had come	14	Incorrect use of infinitive instead of past participle » age truck <i>had come by to emp</i>	had come
to put	15	Incorrect use of participle instead of infinitive » plain how <i>to put recycled</i>	to put
they come	21	Incorrect use of participle instead of simple past or past perfect » <i>uired.</i> <i>So they come in a vari</i>	they came they have come they had come
didn't come	26	Incorrect inflexion of verb after a negated auxiliary » work and <i>didn't come home befo</i>	didn't come

Caution	Graf	Explanation	Suggestion
that	3	that - what follows is essential to the meaning of the sentence which - what follows merely enhances the meaning of the sentence » <i>like said that? Mom? Gr</i>	that which

A Tribute to Spell Catcher

Spell Catcher, a commercial product that has not been supported for some years now after the death of its creator Evan Michael Gross at the end of June 2012, was once an integral part of Literary Toolbox but, as a 32-bit app, it does not run on macOS after Mojave. It was removed in the previous version.

We made our acquaintance with it as Thunder in the 1980s before it became Spell Catcher in 1998. And remain quite fond of it.

So fond, in fact, that we've recreated its Modify menu in **Text Tools**, available separately.

WHAT'S NEW

Literary Toolbox IV is a rewrite of the code in Literary Toolbox III, updating the approach to include subroutines introduced in Keyboard Maestro 10.

At the same time, a few new functions and behavior have been added:

- Searching the online Merriam-Webster dictionary
- Searching the online Merriam-Webster thesaurus
- Displaying online results in a self-contained browser window like the WordNet thesaurus results
- Displaying a Wait window with a progress bar for compute-intensive parsing functions
- Parts of Speech's tagged text displays the formal tag (explained in the accompanying Legend) when you mouse over a colored word
- Addition of Howard Oakley's Nalaprop to Parsing section.
- Addition of hot key triggers for the most common options in each section.
- Both a hot key trigger for the palette and access from the menu bar.

Both the Parts of Speech and Grammarian code has been updated to be a bit more efficient. Parts of Speech has been redesigned a bit too to include scrolling panes for longer texts. The Legend has been cleaned up a little, too.

Palette menu items that used 'Control' or 'Option' to now represent those triggers with keyboard glyphs.

Under the hood, the actions have all been colorized to suggest their function with Execute a Script actions in purple, HTML Prompts in orange and layers of Actions in teal and magenta.

INSTALLATION

Some of the macros require installation of free third-party software.

Nisus Thesaurus

Download [Nisus Thesaurus](#) and install it in your Applications folder.

wnb3 for WordNet

To access a local WordNet database (rather than the online version at Princeton), install the 64-bit [wnb3.app](#), a free 64-bit app by Sergey Sikorskiy in your Applications folder.

Nalaprop

To analyze the selection using macOS's text analysis tools introduced in Catalina, install Howard Oakley's free Nalaprop. Read more about this tool in *Exploring Natural Languages With Nalaprop*.

Lingua::EN::Fathom

Lingua::EN::Opinion

Lingua::EN::Tagger

Lingua::EN::Grammarian

Lingua::EN::Fathom is a free Perl module required for the Word Frequency, Most Frequent Words and Statistics macros. Lingua::EN::Opinion is a free Perl module for the Emotion and Opinion macros. Lingua::EN::Tagger is also a free Perl module required here by the macros in the Parsing section. Lingua::EN::Grammarian, also free, is a Perl module providing grammar cautions and errors.

There are several ways to install Perl modules but here's what we did (and you can do by copying and pasting the following command lines) using the Terminal application:

1. Install cpanm

```
curl -L http://cpanmin.us | perl - --sudo App::cpanminus
```

NB: If you don't have XCode installed, run this command first:

```
xcode-select --install
```

2. Install Lingua::EN::Fathom using cpanm

```
cpanm Lingua::EN::Fathom --sudo
```

3. Install Lingua::EN::Opinion using cpanm

```
cpanm Lingua::EN::Opinion --sudo
```

4. Install Lingua::EN::Tagger using cpanm

```
cpanm Lingua::EN::Tagger --sudo
```

5. Install Lingua::EN::Grammarian

```
cpanm Lingua::EN::Grammarian --sudo
```

You'll also need to download the distribution from cpan at:

<https://metacpan.org/pod/Lingua::EN::Grammarian>

to retrieve the caution and error files, which have to be installed manually.

Copy the grammarian_cautions and grammarian_errors files to any one of the following locations:

```
/usr/local/share/grammarian/
```



```
/Users/[username]/  
/Users/[username]/Desktop/
```

You may hide both files by prepending a period to the file names, if you want.

If you prefer the first solution, open terminal and execute these two commands (after moving the :

```
mkdir /usr/local/share/grammarian
```

```
cp ~/Desktop/grammarian* /usr/local/share/grammarian
```

Notes

The `--sudo` switch will cause the process to ask you for your password. Keep an eye out for it.

Note that the modules have dependencies on other modules which `cpanm` will find and install as well. You need about 60K for all three modules, a modest allocation.

You'll also find `cpanm` helpful for installing updates to any of the modules. Simply rerun the installation command to either check for an update or install any existing update to any particular module.

About Perl

Perl 5 is required. This was traditionally included with macOS but Apple has announced plans to discontinue its inclusion.

If you have installed Perl yourself or multiple versions of Perl, let Keyboard Maestro know where the modules are with a Keyboard Maestro variable named `ENV_PERL5LIB` that includes the path to any Perl modules.

That will avoid Perl errors like a failure to find some module after installing them all according to the documentation.

This only applies to non-standard (custom) macOS Perl installations.

UNINSTALLING MODULES

You can uninstall the Perl modules using `cpanm` (which is another good reason to have installed it):

```
cpanm --uninstall Lingua::EN::Fathom --sudo
```

And if you installed Grammarian, remember to remove the caution and error files you manually installed.

A PALETTE THEME

Keyboard Maestro permits one Custom Palette Theme installed via Terminal. We've written a theme that matches the color scheme of the toolbox reports.

To install the theme paste this command after the prompt in Terminal:

```
defaults write com.stairways.keyboardmaestro.engine  
CustomPaletteTheme -string "0,0,0, 190,72,76, 0,0,100"
```

And in the Keyboard Maestro editor select "Custom" for the palette theme.

This will overwrite any previously installed Custom Palette Theme, so skip it if you've already installed one you rely on.

USAGE

We use these tools from a palette that is active in applications that display text, including editors, email software, browsers, word processors, etc.

We find it useful not only for evaluating what we produce but for measuring the readability of other sources.

The Dictionary and Thesaurus macros work on a word selection. Spell Check Word works on any size selection down to a word. Document spell checks select all the text in the current document. The others will analyze a text selection of any length.

NOTES

There isn't much documentation let alone discussion of the Perl modules that report **emotion** or **opinion** and parse parts of speech. And we've found these tools a bit less accurate than we might have hoped.

But we still find value in them.

For example, while a single word seems to be able to trigger an emotion and other words in the same sentence trigger other emotions, it's helpful to list sentences that may trigger a reaction you might not be expecting.

The same goes for opinions (mainly whether the sentence is positive or negative).

As for **parts of speech**, the task is very difficult but the relevant module seems to do a passable job (you don't have to upgrade to Mojave to get text parsing). The colorization (which is also tonal) helps visualize the logical flow of the text. You can easily perceive if nouns or verbs are modified, the ratio of nouns or noun phrases to verbs and watch for conjunctions and prepositions. In short, you get a painting of your text.

We've lumped together variants of several of the tags. So all verbs have the same color and all nouns have the same color, regardless of variant. Adverbs have the same hue but a different tone than verbs. And adjectives have the same hue but a different tone than nouns.

You can easily change tag colors in the Execute Shell Script action under the section titled `# set colors`. There are three color lists that should (but don't have to) match.

The default arrangement should make it easy to see nouns (red) and verbs (green) and their modifiers (shades of each). Which should stand out from everything else, generally speaking.

Grammarian's cautions list every problematic word it finds whether it is used correctly or not. It just can't tell. When you become familiar with the cautions, you can skip them by invoking Grammarian with the `option-g` trigger.

The **WordNet** options are still being refined. The LT style option may seem superfluous with local and online options available but we find it more readable and it's also what we would use as an interface to a node.js solution to a local version.

RELEASE NOTES

Literary Tools IV

17 August 2022

- Added Merriam-Webster dictionary and thesaurus options
- Changed menu items to use keyboard glyphs
- Parts of Speech rewritten, fully implementing tooltip precise part codes
- Use Keyboard Maestro v10 subroutine function to write temporary file and implement wait window code
- Accessible from the menu bar or via hot key trigger
- Standardize action coloring to match function
- General code review and cleanup

Literary Tools III

19 April 2022

- Replaced global variable SC_WORD with a simple copy and paste of selected text in both the Dictionary and Thesaurus options.

- WordNet: Provides three approaches to Princeton's WordNet database. 'V' by itself activates wnb3.app (a free 64-bit app by Sergey Sikorskiy at <https://www.sikorskiy.net/info/prj/wnb3/>) to look up the selection in a local database. 'Opt-V' retrieves the data from Princeton directly and, like wnb3.app, lets you continue the search in the same window. 'Ctl-V' uses the same dark interface as the rest of Literary Tools but subsequent calls will resort to the Princeton interface.

4 December 2021

- Removed reliance on 32-bit apps Spell Catcher and local WordNet

Literary Tools II

26 October 2021

- Revised Apple Spell Check logic

19 March 2021

- Added local WordNet Lookup option
- Change hot key for WordNet options
- Activated WordNet links in remote option
- Updated Spell Catcher word lookup
- Revised Spell Check Document to include some non-standard keyboard shortcuts which you can extend easily for other applications
- Revised documentation

9 March 2021

- Added WordNet Lookup to access the Princeton database
- Text is written to a temporary file rather than to a Keyboard Maestro maestro variable to accommodate very long documents that would consume too much of the environment as a variable.
- Updated code to handle Updated code to handle the revised Emotion module.
- Removed restrictions on InDesign CS5 as obsolete.

Literary Tools (the original)

10 November 2020

- Two-column reports for Most Used Words and Word Frequency

- New method of calling Spell Catcher (thanks to Chris Stone)
- Revised documentation updating Custom Theme code (thanks to slimhan)

19 July 2020

- Added context to Grammarian listings
- Revised HTML window CSS

17 March 2019

- Most Used Words option updated with more precise regexp

13 December 2018

- Manual updated with installation tips.

3 December 2018

- Initial release.

CONTACT

I'm happy to promptly address any concerns you may have. I can't do anything about the modules themselves but I can address interface issues.

You can reach me at <http://mikepasini.com> or on the Keyboard Maestro forum as mrpasini.