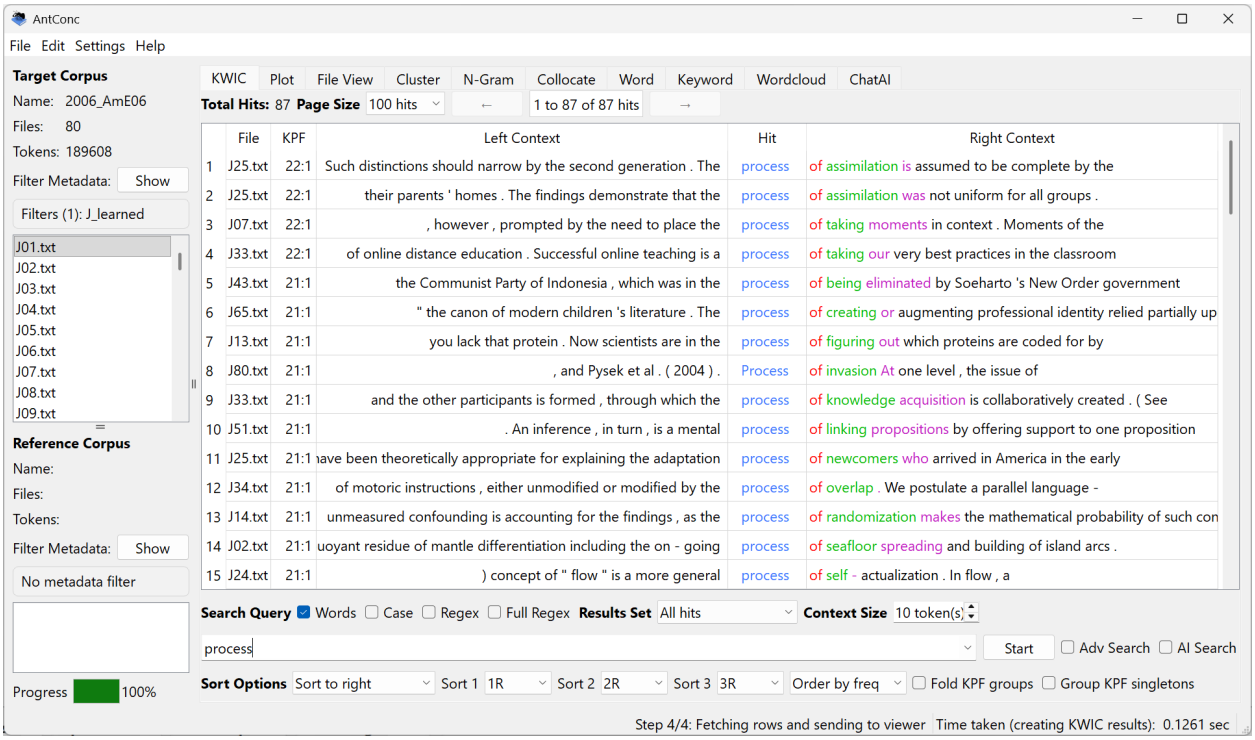
AntConc

Laurence Anthony, Ph.D.
Faculty of Science and Engineering,
Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan
April 14, 2026

Table of Contents

Introduction



AntConc is freeware, cross-platform software for corpus linguistics research, corpus-based teaching, and data-driven language learning. It runs on Microsoft Windows, macOS, and Linux.

AntConc is developed in Python with Qt and packaged for each operating system using PyInstaller. Corpus data and tool results are managed with SQLite.

Getting Started

 **Windows (Installer)**

Double-click the *AntConc installer package* and follow the installer instructions. After installation, you can launch AntConc from the Start menu.

 **Windows (Portable)**

Unzip the *AntConc portable ZIP package* into a folder of your choice. In the *AntConc* folder, double-click the *AntConc.exe* file to launch the program.

 **macOS**

Double-click the *AntConc disk image* to mount it. Open the disk image and drag the *AntConc* app to the Applications folder (or another location if you prefer). You can then launch the app from Applications or Launchpad.

 **Linux**

Download the *AntConc Flatpak bundle* and install it with `flatpak install <bundle-file>`. After installation, you can launch *AntConc* from your desktop application menu or from the command line with `flatpak run org.antconc.AntConc`.

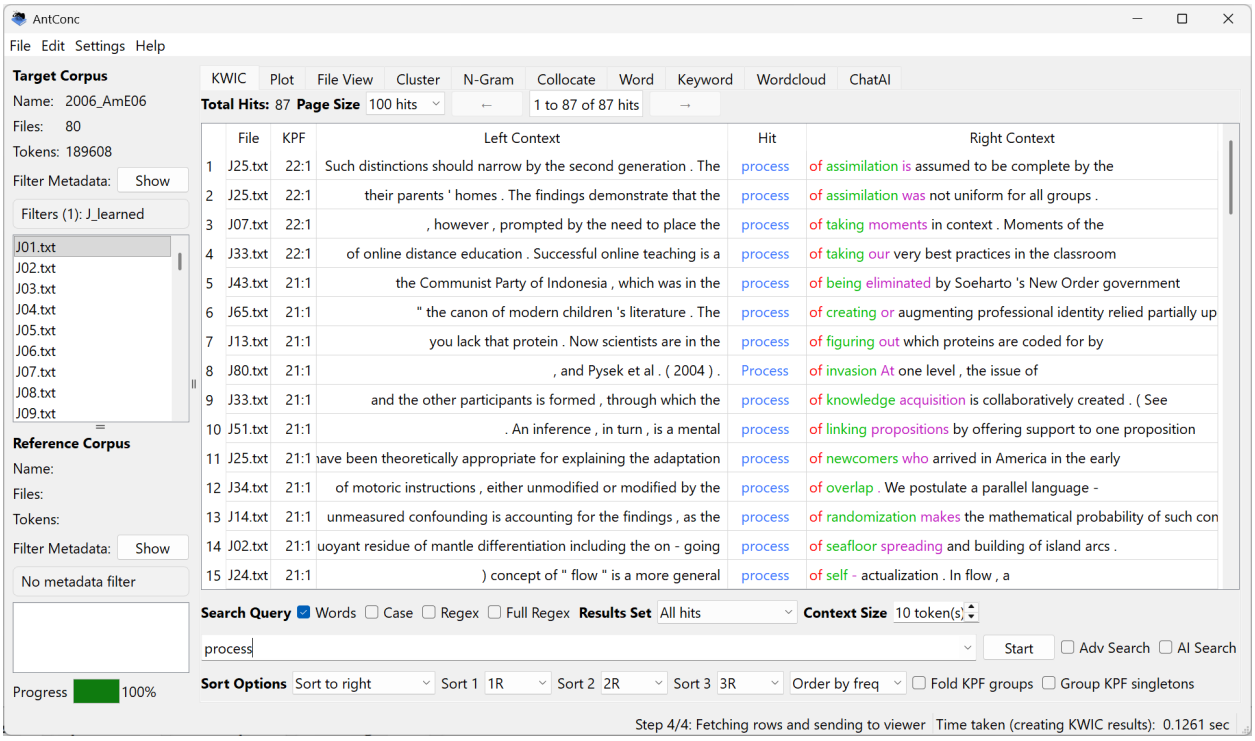
Overview of Tools

AntConc contains ten tools that can be accessed either by clicking on their tabs in the tool window or by using shortcut keys (see the `Shortcuts` section for the current key mappings).

- KWIC (Key-Word-In-Context) Tool
  - Displays concordance lines in Key Word in Context format so you can inspect usage patterns in context.
- Plot Tool
  - Displays concordance hits as barcode-style distribution plots so you can see where matches occur across texts.
- File View Tool
  - Displays full source texts so you can inspect hits and surrounding context in detail.
- Cluster Tool
  - Finds adjacent word sequences around a query so you can identify frequent phrase patterns.
- N-Gram Tool
  - Finds all frequent n-word sequences in the corpus, optionally with open slots, to reveal recurring expressions.
- Collocate Tool
  - Finds words that co-occur near a query term within a defined span to identify association patterns.
- Word List Tool
  - Generates frequency lists of words in the corpus to identify common vocabulary.
- Keyword List Tool
  - Compares target and reference corpora to identify statistically distinctive words (including negative keywords).

- Wordcloud Tool
  - Visualizes results from other tools (or scratchpad text) as a word cloud for quick pattern overview.
- ChatAI
  - Lets you interact with LLMs directly or with AntConc tool output as context for guided analysis.

KWIC ('Key-Word-In-Context') Tool



The KWIC tool generates concordance lines for a search query and presents each hit with left and right context. It is designed for fast pattern discovery, close reading of usage, and easy movement from summary results to full-text evidence in the File View tool.

Practical Examples

- Research: Search for a lemma or phrase (e.g., *sustainability*) and sort by surrounding words to identify recurring lexical patterns and phraseology in a corpus.
- Teaching: Build a classroom activity where students infer the missing center word from KWIC lines, then reveal the hit to discuss grammar, collocation, and register.
- Student learning: Compare near-synonyms (e.g., *big* vs *large*) by running separate KWIC searches and inspecting differences in typical contexts.

Core Workflow

1) Load a target corpus via File -> Open Corpus Manager, or use File -> Open File(s) as Quick Corpus.
2) Enter a query in the KWIC search box.
3) Set the search mode as needed: "Words", "Case", "Regex", or "Full Regex" (if available).
   - "Words": Match whole words/tokens rather than partial substrings.

- "Case": Make matching case-sensitive.
- "Regex": Use regular expression patterns at the token level (matching within tokenized word units).
- "Full Regex" (if available): Use regular expression matching at the document level (across full text spans); this mode disables "Words" and "Adv Search" for that run.

4) (Optional) Adjust "Results Set" and "Context Size". Defaults are suitable for a first run.
5) Click "Start" (or press Enter in the query box) to generate KWIC lines.
6) Read the "Total Hits" value and inspect the concordance lines in the results table.

Additional Features

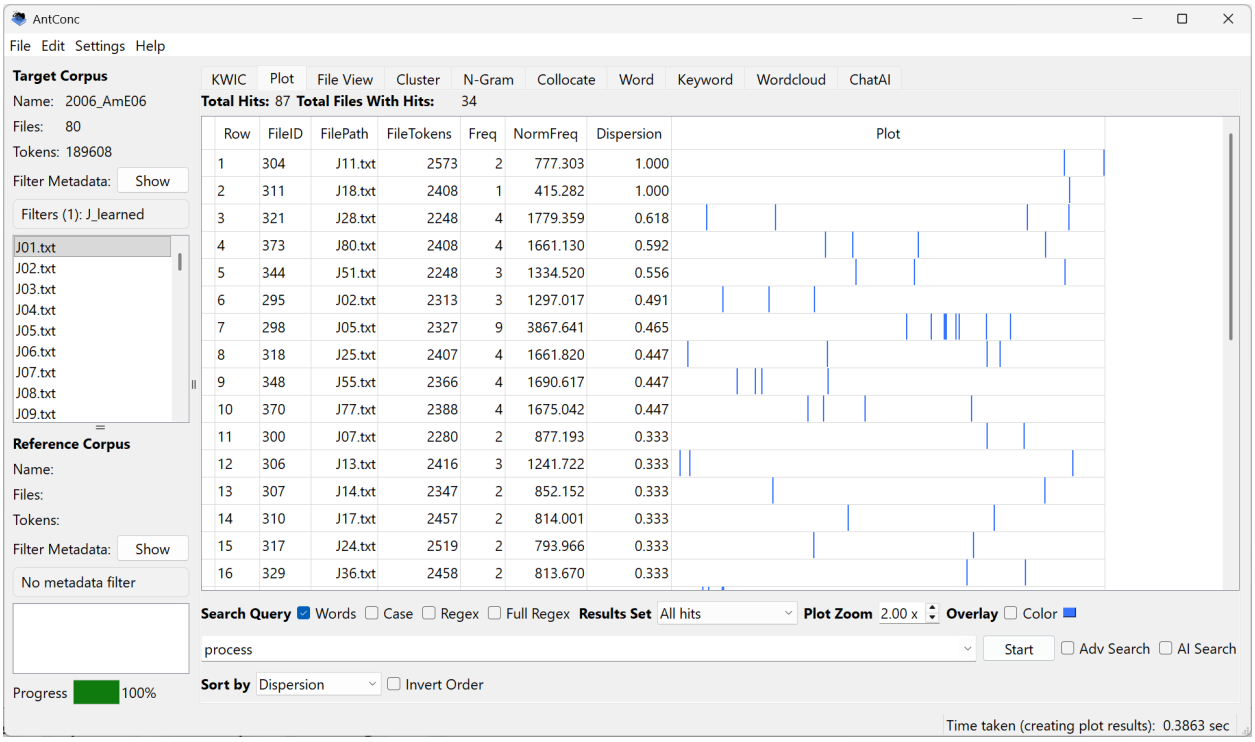
- Sort controls: Use "Sort Options" to choose a preset order (e.g., sort to right/left, center-to-right/left) or "Custom" order. You can set three sort keys (1L/2L/.../C/.../File/File ID/Row ID) and sort by "Order by value" or "Order by freq".
- Results subset and paging: "Results Set" can show all hits or a random subset. When a random subset is chosen, pagination controls are hidden.
- Pagination: Use the page-size and page controls above the table to browse large result sets.
- KPF (KWIC Pattern Frequency): The KPF column displays values in the form `Total Score: Count` or `Total Score: Count (Group Size)`. The first value is the total pattern score. The second value is the number of rows with the same KPF score shown together. If `Group singletons` is enabled, the number in brackets shows the total number of singleton (single pattern) rows grouped together.
- Jump to evidence: Double-click a KWIC row to open the corresponding file context in the File View tool.
- Row filtering shortcuts: Select rows and press Delete to remove selected rows from the current KWIC view; press Shift+Delete to keep selected rows and remove others.
- Advanced Search: Enable "Adv Search" to open list/context/SQL-style constraints (where supported).
- AI Search: Enable "AI Search" to transform natural-language prompts into search terms before running KWIC.
- Validation checks: KWIC warns when no corpus is loaded, no query is entered, or the query span exceeds the current context size.

Tool Settings

Open via Settings -> Tool Settings -> KWIC Settings.

- Display Type
  - Select how tokens are shown in KWIC (for example, type/POS/headword combinations available for the current corpus/indexer).
- Display Options
  - Hide file names in display.
  - Hide search term in display (useful for allowing instructors to quiz students on possible words to fit the gap).
  - Set file name column width.
- Display Color Options
  - Set Level 1, Level 2, and Level 3 sort colors used in the KWIC table.
- AI System Prompt
  - Edit the system prompt used when "AI Search" is enabled. This guides the LLM during interactions. It can be used to strengthen model behavior (e.g., "You are an expert corpus linguist"), set interaction style (e.g., "Use a formal style"), and specify how it should proceed (e.g., "Analyze the data in a stepwise fashion").

Plot Tool



The Plot tool visualizes where search hits occur across files by drawing hit positions along each document timeline. It is designed for distribution analysis: you can quickly compare concentration, spread, and relative placement of hits across documents, then jump from a plotted hit to its source context in the File View tool.

Practical Examples

- Research: Compare whether discipline-specific terms are evenly distributed across articles or concentrated in specific sections/documents.
- Teaching: Show how a discourse marker (for example, *however*) is distributed through model essays, then discuss rhetorical structure.
- Student learning: Run two separate searches for near-synonyms and overlay plots to compare where each tends to appear in texts.

Core Workflow

1) Load a target corpus via File -> Open Corpus Manager, or use File -> Open File(s) as Quick Corpus.
2) Enter a query in the Plot search box.
3) Set the search mode as needed: “Words”, “Case”, “Regex”, or “Full Regex” (if available).
    - “Words”: Match whole words/tokens rather than partial substrings.
    - “Case”: Make matching case-sensitive.
    - “Regex”: Use regular expression patterns at the token level (matching within tokenized word units).
    - “Full Regex” (if available): Use regular expression matching at the document level (across full text spans); this mode disables “Words” and “Adv Search” for that run.
4) (Optional) Adjust “Plot Zoom” for display scale. Defaults are suitable for a first run.
5) Click “Start” (or press Enter in the query box) to generate plots.

6) Read “Total Hits” and “Total Files With Hits”, then inspect the plot rows.

Additional Features

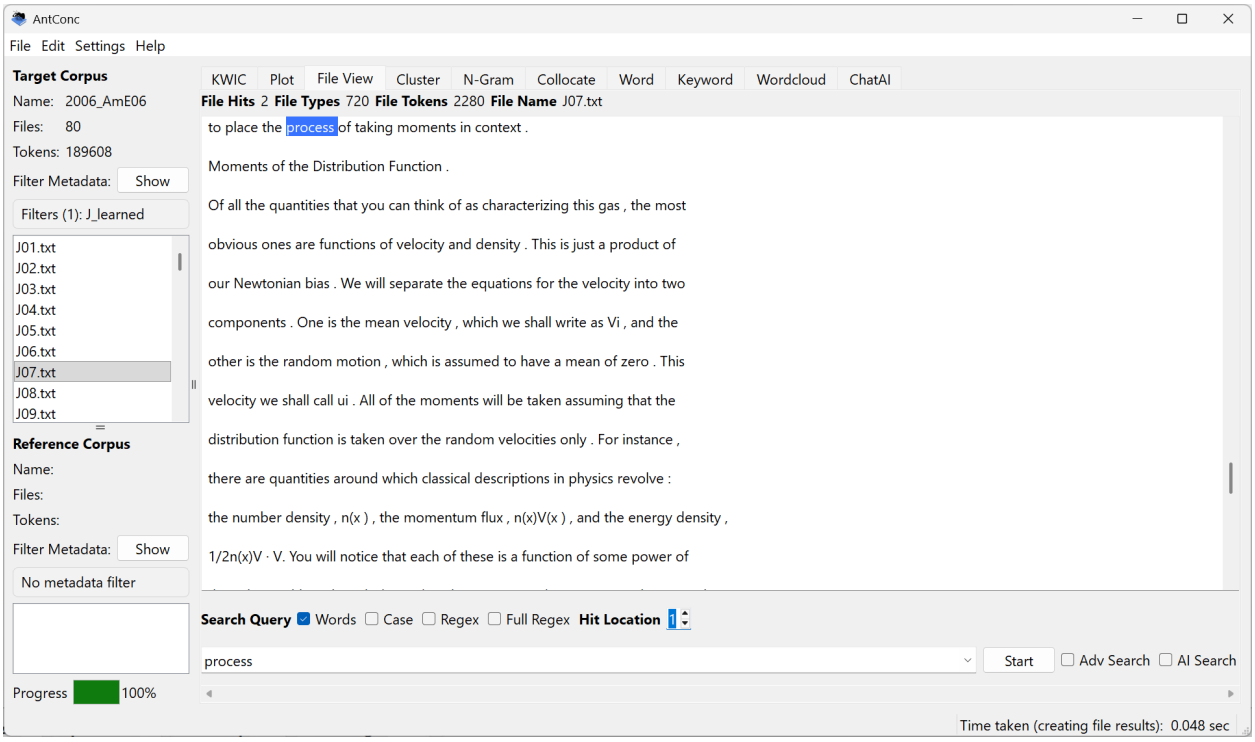
- Sort and order: Use “Sort by” (DocID, DocPath, DocTokens, Frequency, NormFrequency, Dispersion) and “Invert Order” to reorder plot rows.
- Overlay comparisons: Enable “Overlay” and choose a color to draw additional query results on top of existing plots for comparison.
- Jump to evidence: Click a plotted hit line (when plots are displayed) to jump to the corresponding location in File View.
- Advanced Search: Enable “Adv Search” to apply list/context/SQL-style constraints (when supported).
- AI Search: Enable “AI Search” to convert natural-language prompts into search terms before plotting.
- Validation checks: Plot warns when a search is already running, when no corpus is loaded, when no query is entered, or when no hits are found.

Tool Settings

Open via Settings -> Tool Settings -> Plot Settings.

- View Style
  - Table view: Shows a row-based table of dispersion statistics (for example, frequency, normalized frequency, and dispersion) together with an inline plot for each file.
  - Graphic view: Prioritizes the plot visualization over the table layout; useful when focusing on visual comparison and exporting the graphic.
- Display Options
  - Show plots.
  - Show hit ids.
  - Show files with no hits.
  - Normalize plots: When enabled, each document bar is shown at a normalized length so relative hit positions can be compared directly across files. When disabled, each document is shown at its relative length.
- Statistics
  - Dispersion bin size.
  - Dispersion measure (Juilland’s D, Range, Standard Deviation, Standard Deviation (Normed)).
- Other Options
  - Normalized plot width.
  - Relative plot width scalar.
  - Table view plot height.
  - Plot line width.
- AI Settings
  - Edit the system prompt used when “AI Search” is enabled. This will guide the LLM during an interaction (see the KWIC tool settings for details).

File View Tool



The File View tool displays the full text of a selected file and highlights search hits directly in that text. It is designed for close reading: you can inspect local context, move quickly between hits, and pass selected hit text to other tools (for example, KWIC) for broader corpus comparison.

Practical Examples

- Research: Verify how a term is used in its full document context after finding it in KWIC or Plot.
- Teaching: Walk students through sentence-level evidence in one file and discuss why specific tokens are highlighted.
- Student learning: Check whether your target expression appears naturally in authentic texts, then jump between hits to compare usage.

Core Workflow

1) Load a target corpus via File -> Open Corpus Manager, or use File -> Open File(s) as Quick Corpus.
2) Select a file in the "Target Corpus" list (double-clicking a file also opens it in File View).
3) (Optional) Enter a query to highlight hits in the selected file. If left blank, File View still loads the file text without hit highlights.
4) Set the search mode as needed: "Words", "Case", "Regex", or "Full Regex" (if available).
    - "Words": Match whole words/tokens rather than partial substrings.
    - "Case": Make matching case-sensitive.
    - "Regex": Use regular expression patterns at the token level (matching within tokenized word units).
    - "Full Regex" (if available): Use regular expression matching at the document level (across full text spans); this mode disables "Words" and "Adv Search" for that run.
5) Click "Start" (or press Enter in the query box) to load the file and apply highlighting.

6) Use “Hit Location” to step through hits in the file; the selected hit is emphasized in the text.

Additional Features

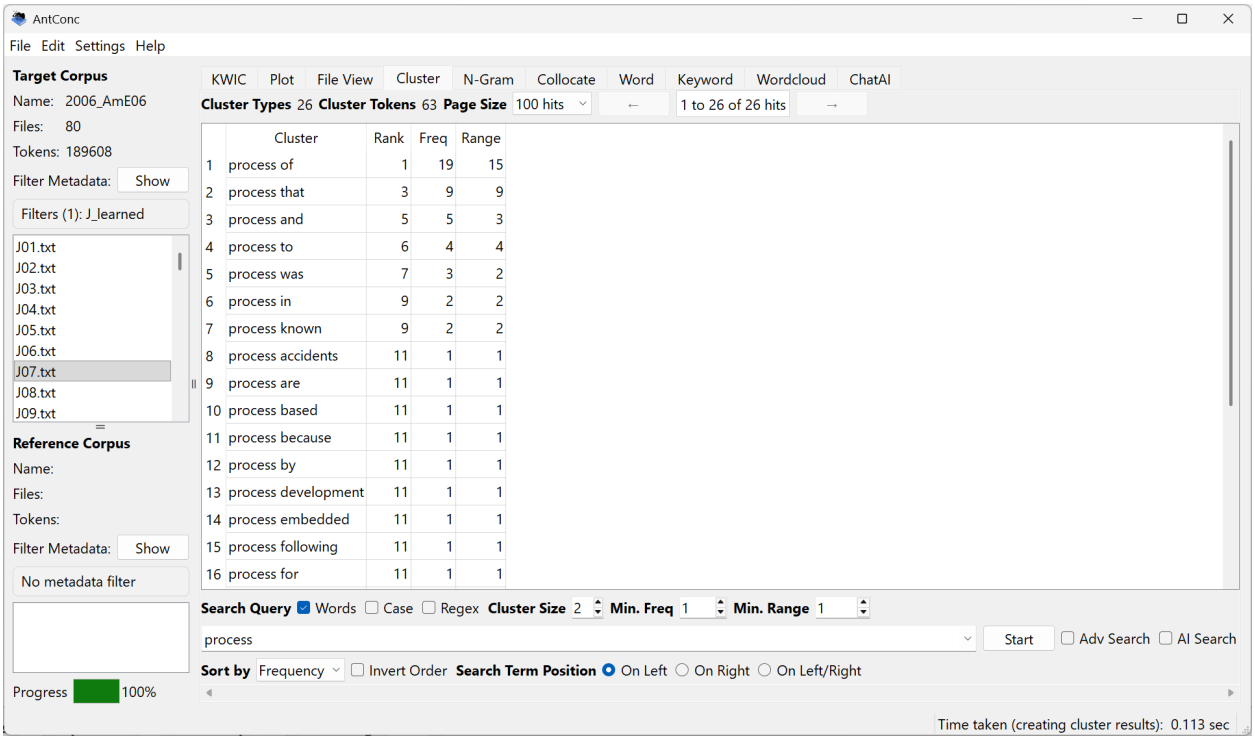
- File statistics header: File View shows “File Hits”, “File Types”, “File Tokens”, and “File Name” for the selected document.
- Jump to KWIC from file text: Double-click a highlighted hit in File View to send that hit text to KWIC and run a concordance search.
- Hit navigation shortcuts: Use Ctrl+F to move forward through hits and Ctrl+Shift+F to move backward.
- Advanced Search: Enable “Adv Search” to apply list/context/SQL-style constraints (when supported).
- AI Search: Enable “AI Search” to convert natural-language prompts into search terms before search/highlighting.
- Validation checks: File View warns when a search is already running, when no corpus is loaded, or when no file is selected.

Tool Settings

Open via Settings -> Tool Settings -> File View Settings.

- Display Type
  - Choose which annotation layer is displayed in the file text (for example, type, POS, headword, and available combinations) based on the current corpus/indexer.
- AI Settings
  - Edit the system prompt used when “AI Search” is enabled. This will guide the LLM during an interaction (see the KWIC tool settings for details).

Cluster Tool



The Cluster tool finds recurring multi-word sequences around a search term and summarizes them as ranked cluster types. It is designed to reveal phrase-level patterning in a corpus, including which combinations are frequent, widespread across files, and worth inspecting further in KWIC.

Practical Examples

- Research: Identify recurring phrase frames around a keyword (for example, *in terms of*) and compare how often they recur across documents.
- Teaching: Show learners typical phrase extensions before or after a target word to discuss chunking and formulaic language.
- Student learning: Generate clusters for a key term and use the ranked output to learn common phrase patterns for writing.

Core Workflow

1) Load a target corpus via File -> Open Corpus Manager, or use File -> Open File(s) as Quick Corpus.
2) Enter a search query in the Cluster search box.
3) Set the search mode as needed: "Words", "Case", and/or "Regex".
    - "Words": Match whole words/tokens rather than partial substrings.
    - "Case": Make matching case-sensitive.
    - "Regex": Use regular expression patterns at the token level (matching within tokenized word units).
4) Set the main cluster parameters: "Cluster Size", "Min. Freq", and "Min. Range".
5) Choose "Search Term Position" ("On Left", "On Right", or "On Left/Right") based on where the query should appear in each cluster.
6) Click "Start" (or press Enter in the query box) to generate cluster results.
7) Read "Cluster Types" and "Cluster Tokens", then inspect the ranked rows in the table.

Additional Features

- Sort and order: Use "Sort by" (Type, TypeEnd, Frequency, Range) and "Invert Order" to reorder results.
- Jump to KWIC: Double-click a cluster in the "Cluster" column to open KWIC results for that cluster string.
- Pagination: Use the page-size and page controls above the table to browse large result sets.
- Advanced Search: Enable "Adv Search" to apply list/context/SQL-style constraints (when supported).
- AI Search: Enable "AI Search" to convert natural-language prompts into search terms before clustering.
- Validation checks: Cluster warns when a search is already running, when no corpus is loaded, when no query is entered, or when query span exceeds "Cluster Size".

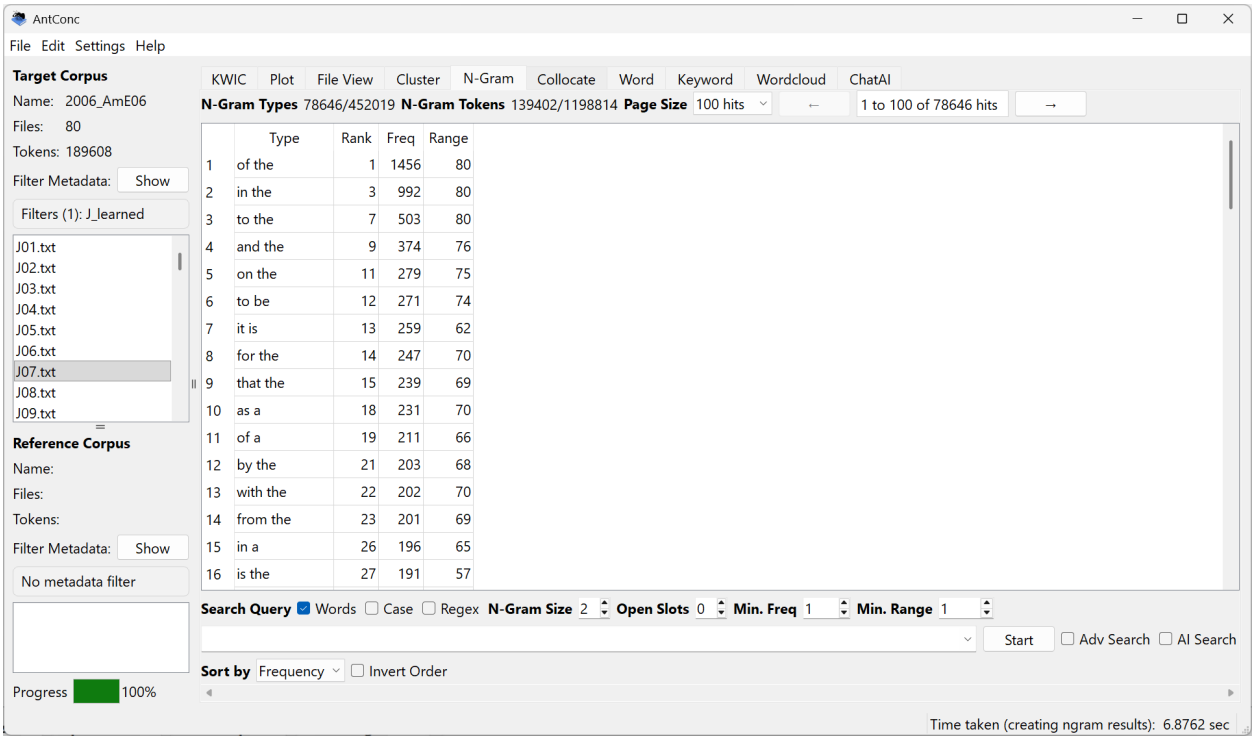
Tool Settings

Open via Settings -> Tool Settings -> Cluster Settings.

- Display Type
    - Choose which annotation layer is used to build/display clusters (for example, type, POS, headword, and available combinations) based on the current corpus/indexer.
- Display Options
    - Show raw values.

  - Show normalized values.
- Filter Options
  - Only include clusters with whitespace separators.
- AI Settings
  - Edit the system prompt used when "AI Search" is enabled. This will guide the LLM during an interaction (see the KWIC tool settings for details).

N-Gram Tool



The N-Gram tool extracts recurring multi-word sequences of a specified length and ranks them by frequency and range. It is designed to identify common phrase patterns in a corpus and, when open slots are enabled, to examine patterned variation inside those sequences.

Practical Examples

- Research: Identify high-frequency lexical bundles (for example, 3-grams or 4-grams) and compare how widely they occur across files.
- Teaching: Build phrase-list activities from common n-grams to teach chunking and formulaic language.
- Student learning: Explore typical multi-word patterns in a target genre and reuse them for more natural writing.

Core Workflow

1) Load a target corpus via File -> Open Corpus Manager, or use File -> Open File(s) as Quick Corpus.
2) Set the main n-gram parameters: "N-Gram Size", "Open Slots", "Min. Freq", and "Min. Range".
3) (Optional) Enter a search query to filter n-grams.
4) Set the search mode as needed: "Words", "Case", and/or "Regex".

- "Words": Match whole words/tokens rather than partial substrings.
- "Case": Make matching case-sensitive.
- "Regex": Use regular expression patterns at the token level (matching within tokenized word units).

5) Click "Start" (or press Enter in the query box) to generate n-gram results.
6) Read "N-Gram Types" and "N-Gram Tokens", then inspect rows in the results table.

Additional Features

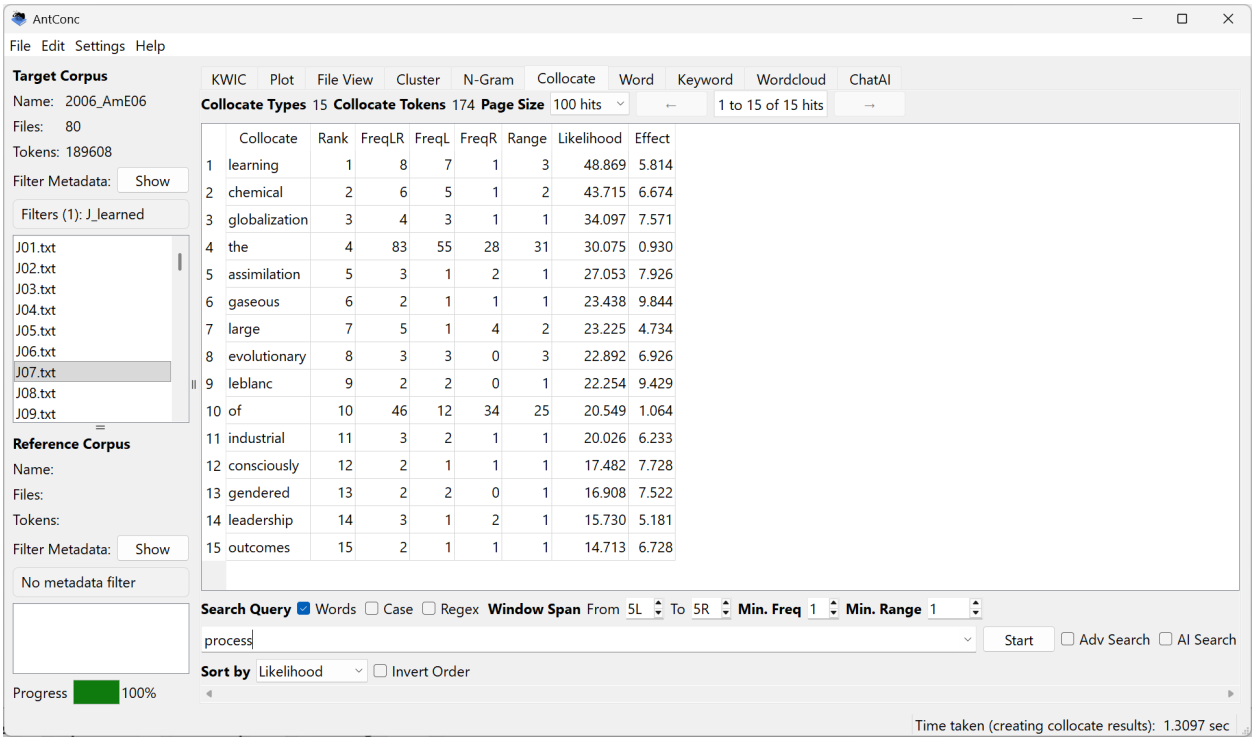
- Sort and order: Use "Sort by" (Type, TypeEnd, Frequency, Range, plus slot metrics when available) and "Invert Order" to reorder rows.
- Open-slot metrics: When "Open Slots" is greater than 0, additional columns (for example, `S1_TT` and `S1_Ent`) are shown for type/token ratio and entropy by slot.
- Open Slot Viewer: Shift+double-click a "Type" cell (with open slots enabled) to open the slot-variant viewer for that n-gram.
- Jump to KWIC: Double-click an n-gram in the "Type" column to open KWIC results for that string.
- Pagination: Use the page-size and page controls above the table to browse large result sets.
- Advanced Search: Enable "Adv Search" for list-based query constraints.
- AI Search: Enable "AI Search" to convert natural-language prompts into search terms before n-gram search.
- Validation checks: N-Gram warns when a search is already running, when no corpus is loaded, or when query span exceeds "N-Gram Size".

Tool Settings

Open via Settings -> Tool Settings -> N-Gram Settings.

- Display Type
  - Choose which annotation layer is used to build/display n-grams (for example, type, POS, head-word, and available combinations) based on the current corpus/indexer.
- Display Options
  - Show raw values.
  - Show normalized values.
- Filter Options
  - Only include ngrams with whitespace separators.
- AI Settings
  - Edit the system prompt used when "AI Search" is enabled. This will guide the LLM during an interaction (see the KWIC tool settings for details).

Collocate Tool



	Collocate	Rank	FreqLR	FreqL	FreqR	Range	Likelihood	Effect
1	learning	1	8	7	1	3	48.869	5.814
2	chemical	2	6	5	1	2	43.715	6.674
3	globalization	3	4	3	1	1	34.097	7.571
4	the	4	83	55	28	31	30.075	0.930
5	assimilation	5	3	1	2	1	27.053	7.926
6	gaseous	6	2	1	1	1	23.438	9.844
7	large	7	5	1	4	2	23.225	4.734
8	evolutionary	8	3	3	0	3	22.892	6.926
9	leblanc	9	2	2	0	1	22.254	9.429
10	of	10	46	12	34	25	20.549	1.064
11	industrial	11	3	2	1	1	20.026	6.233
12	consciously	12	2	1	1	1	17.482	7.728
13	gendered	13	2	2	0	1	16.908	7.522
14	leadership	14	3	1	2	1	15.730	5.181
15	outcomes	15	2	1	1	1	14.713	6.728

The Collocate tool identifies words that co-occur with a search term within a defined left/right window span. It is designed for co-occurrence analysis: you can compare directional frequencies, range, and association strength (likelihood/effect) to find meaningful collocational patterns.

Practical Examples

- Research: Compare which words most strongly co-occur with a target term (for example, *climate*) and whether they prefer left or right positions.
- Teaching: Demonstrate how common collocates around a verb differ by grammatical pattern and position.
- Student learning: Build a collocation list for a key topic word, then check each item in KWIC for contextual confirmation.

Core Workflow

1) Load a target corpus via File -> Open Corpus Manager, or use File -> Open File(s) as Quick Corpus.
2) Enter a search query in the Collocate search box.
3) Set the search mode as needed: “Words”, “Case”, and/or “Regex”.
    - “Words”: Match whole words/tokens rather than partial substrings.
    - “Case”: Make matching case-sensitive.
    - “Regex”: Use regular expression patterns at the token level (matching within tokenized word units).
4) Set “Window Span” (From/To), “Min. Freq”, and “Min. Range”.
5) Click “Start” (or press Enter in the query box) to generate collocate results.
6) Read “Collocate Types” and “Collocate Tokens”, then inspect rows in the results table.

Additional Features

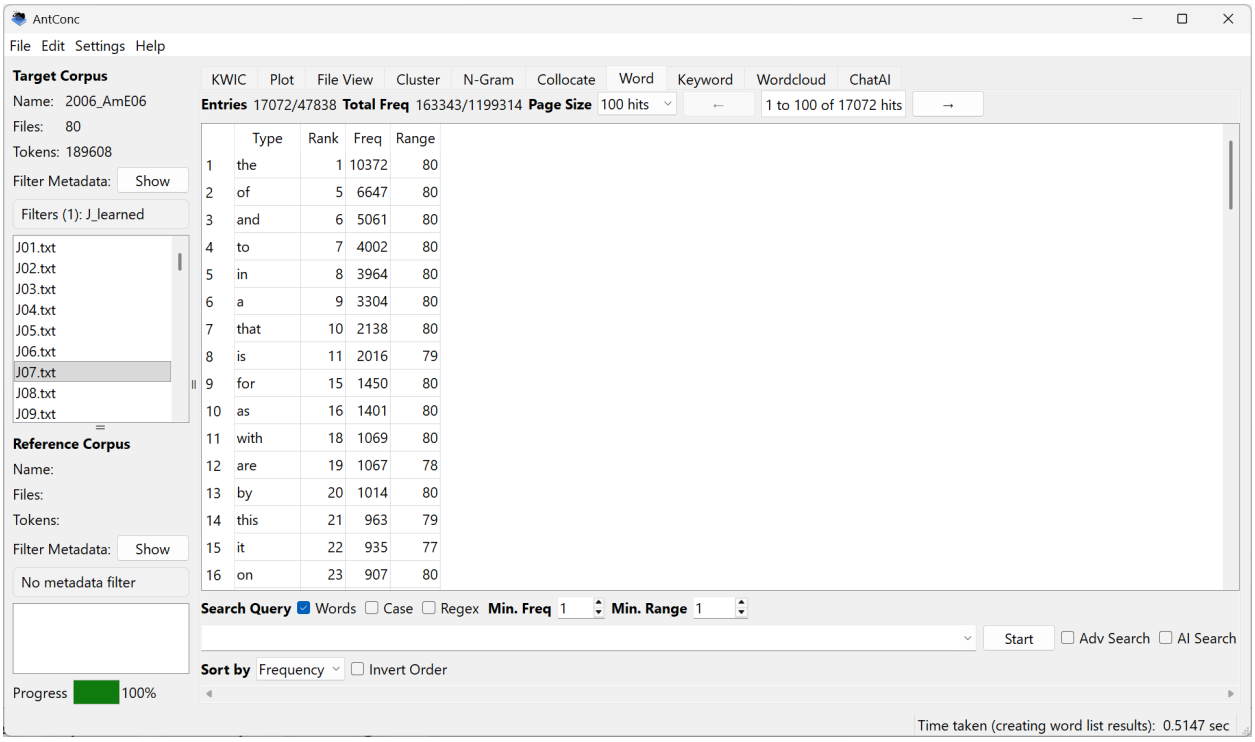
- Sort and order: Use "Sort by" (Type, TypeEnd, Frequency(LR), Frequency(L), Frequency(R), Range, Likelihood, Effect) and "Invert Order" to reorder results.
- Directional frequency columns: Compare `FreqLR`, `FreqL`, and `FreqR` to see overall and side-specific collocate behavior.
- Jump to KWIC: Double-click a collocate in the "Collocate" column to open KWIC results for that collocate/query combination.
- Pagination: Use the page-size and page controls above the table to browse large result sets.
- Advanced Search: Enable "Adv Search" to apply list/context/SQL-style constraints (when supported).
- AI Search: Enable "AI Search" to convert natural-language prompts into search terms before collocate search.
- Validation checks: Collocate warns when a search is already running, when no corpus is loaded, or when no query is entered.

Tool Settings

Open via Settings -> Tool Settings -> Collocate Settings.

- Display Type
  - Choose which annotation layer is used to build/display collocates (for example, type, POS, headword, and available combinations) based on the current corpus/indexer.
- Likelihood Measure
  - Select the likelihood statistic (for example, Log-Likelihood 4-term or 2-term).
  - Set the likelihood threshold (for example, `All values`, `p<0.05`, `p<0.01`, with optional Bonferroni variants).
- Effect Size Measure
  - Select the effect-size statistic (for example, MI, MI2, MI3, Dice, LogDice, LogRatio, T-Score, Z-Score).
  - Set the effect threshold (for example, `All values`, `top 10`, `top 50`, ...).
- AI Settings
  - Edit the system prompt used when "AI Search" is enabled. This will guide the LLM during an interaction (see the KWIC tool settings for details).

Word Tool



The Word tool generates a frequency list of word entries from the target corpus and ranks them by frequency, range, or lexical form. It is designed for lexical profiling: you can inspect how often items occur, how widely they are distributed across files, and quickly pivot to KWIC for context.

Practical Examples

- Research: Identify the most frequent content words in a corpus and compare frequency/range patterns for candidate keywords.
- Teaching: Build frequency-based vocabulary lists for a course unit and discuss which items are broadly distributed versus text-specific.
- Student learning: Check common words and forms in a genre, then open KWIC lines to see typical usage in context.

Core Workflow

1) Load a target corpus via File -> Open Corpus Manager, or use File -> Open File(s) as Quick Corpus.
2) Enter a query only if you want to filter the word list to matching entries; otherwise leave it blank for a full word list. Set the search mode as needed: "Words", "Case", and/or "Regex".
    - "Words": Match whole words/tokens rather than partial substrings.
    - "Case": Make matching case-sensitive.
    - "Regex": Use regular expression patterns at the token level (matching within tokenized word units).
3) Set "Min. Freq" and "Min. Range".
4) Click "Start" (or press Enter in the query box) to generate results.
5) Read the summary counts at the top (entries/total frequency) and inspect ranked rows in the table.

Additional Features

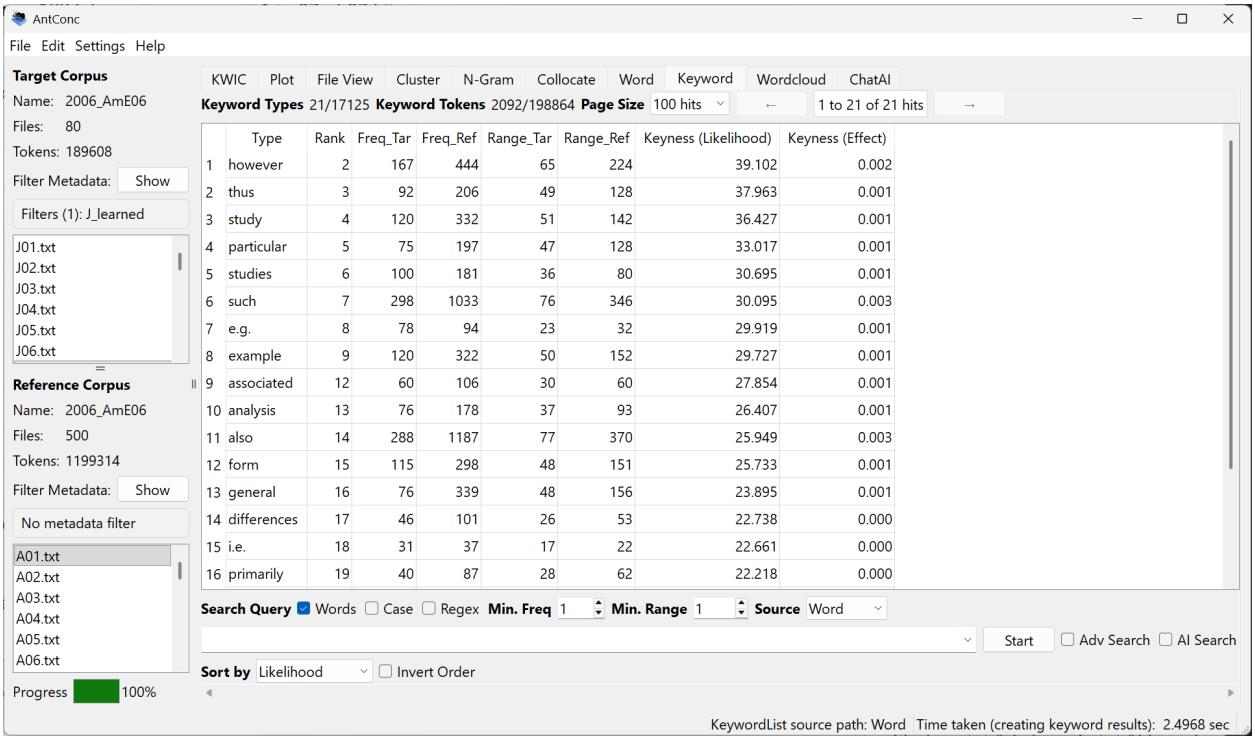
- Sort and order: Use "Sort by" (dynamic lexical fields plus Frequency/Range) and "Invert Order" to reorder rows.
- Jump to KWIC: Double-click a word entry in the first column to open KWIC results for that item.
- Pagination: Use the page-size and page controls above the table to browse large result sets.
- Advanced Search: Enable "Adv Search" to apply list/context/SQL-style constraints (when supported for the current corpus format).
- AI Search: Enable "AI Search" to convert natural-language prompts into search terms before word-list search.
- Validation checks: Word tool warns when a search is already running or when no corpus is loaded.

Tool Settings

Open via Settings -> Tool Settings -> Word List Settings.

- Display Type
  - Choose which annotation layer is used to build/display the word list (for example, type, POS, headword, and available combinations) based on the current corpus/indexer.
- Display Options
  - Show raw values.
  - Show normalized values.
- AI Settings
  - Edit the system prompt used when "AI Search" is enabled. This will guide the LLM during an interaction (see the KWIC tool settings for details).

Keyword Tool



	Type	Rank	Freq_Tar	Freq_Ref	Range_Tar	Range_Ref	Keyness (Likelihood)	Keyness (Effect)
1	however	2	167	444	65	224	39.102	0.002
2	thus	3	92	206	49	128	37.963	0.001
3	study	4	120	332	51	142	36.427	0.001
4	particular	5	75	197	47	128	33.017	0.001
5	studies	6	100	181	36	80	30.695	0.001
6	such	7	298	1033	76	346	30.095	0.003
7	e.g.	8	78	94	23	32	29.919	0.001
8	example	9	120	322	50	152	29.727	0.001
9	associated	12	60	106	30	60	27.854	0.001
10	analysis	13	76	178	37	93	26.407	0.001
11	also	14	288	1187	77	370	25.949	0.003
12	form	15	115	298	48	151	25.733	0.001
13	general	16	76	339	48	156	23.895	0.001
14	differences	17	46	101	26	53	22.738	0.000
15	i.e.	18	31	37	17	22	22.661	0.000
16	primarily	19	40	87	28	62	22.218	0.000

The Keyword tool compares a target corpus against a reference corpus to identify items that are unusually frequent (or, optionally, unusually infrequent) in the target corpus. Depending on the selected `Source`, it can be used to generate keywords for words, clusters, n-grams, or collocates. It is designed for keyness analysis using likelihood and effect-size statistics, with sortable results and quick jump-out to KWIC for contextual checking.

Practical Examples

- Research: Compare two corpora (for example, learner vs reference writing) to identify terms that are statistically overused in the target set.
- Teaching: Show students which words are characteristic of a genre by comparing course texts against a general-language reference corpus.
- Student learning: Build a keyword list for your assignment corpus, then inspect key items in KWIC to verify meaning and usage.

Core Workflow

1) Load a target corpus and a reference corpus via File -> Corpus Manager.
2) Select a `Source` (`Word`, `Cluster`, `N-Gram`, or `Collocate`) to choose what kind of items will be compared for keyness.
3) Enter a query only if you want to filter the keyword analysis to matching entries; otherwise leave it blank for a full keyword list. Set search mode as needed: "Words", "Case", and/or "Regex".
    - "Words": Match whole words/tokens rather than partial substrings.
    - "Case": Make matching case-sensitive.
    - "Regex": Use regular expression patterns at the token level (matching within tokenized word units).
4) Set "Min. Freq" and "Min. Range", then click "Start" (or press Enter in the query box).
5) Read "Keyword Types" and "Keyword Tokens", then inspect ranked rows in the table.

Additional Features

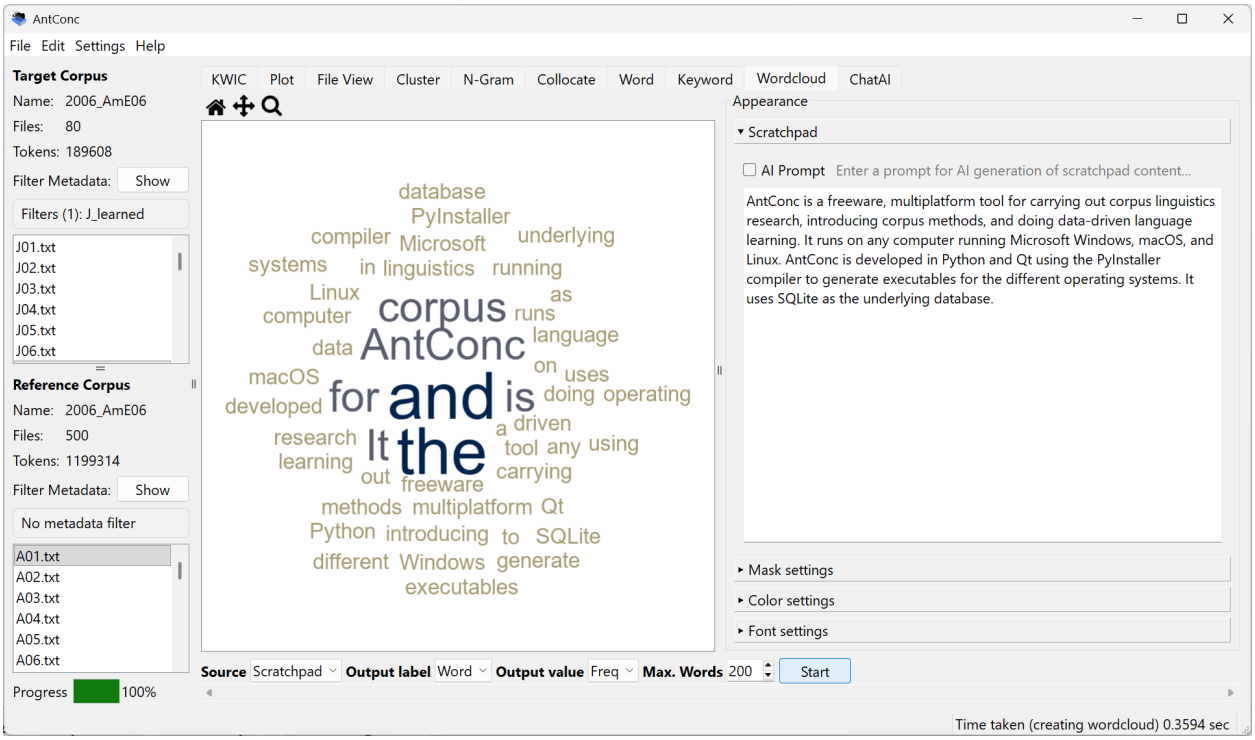
- Sort and order: Use "Sort by" (Type, TypeEnd, Frequency(Tar), Range(Tar), Frequency(Ref), Range(Ref), Likelihood, Effect) and "Invert Order".
- Comparative columns: Inspect target/reference frequencies and ranges side by side, with optional normalized columns.
- Jump to KWIC: Double-click a keyword in the first column to open KWIC results for that entry.
- Pagination: Use the page-size and page controls above the table to browse large result sets.
- Advanced Search: Enable "Adv Search" to apply list/context/SQL-style constraints to target-side query filtering.
- AI Search: Enable "AI Search" to convert natural-language prompts into search terms before keyword search.
- Validation checks: Keyword warns when a search is already running, when target/reference corpora are missing, or when the corpora are not comparable for list type.

Tool Settings

Open via Settings -> Tool Settings -> Keyword List Settings.

- Display Type
  - Choose which annotation layer is used for keyword comparison (for example, type, POS, headword, and available combinations) based on overlap between target/reference corpora.
- Display Options
  - Show raw values.
  - Show normalized values.
- Negative Keywords
  - Show negative keywords (items underrepresented in the target corpus relative to the reference corpus).
- Likelihood Measure
  - Select the keyness statistic (for example, Chi-Squared, Log-Likelihood, TextDispersionKeyness).
  - Set the likelihood threshold (for example, `All values`, `p<0.05`, `p<0.01`, with optional Bonferroni variants).
- Effect Size Measure
  - Select the effect-size statistic (for example, Dice, LogDice, LogRatio, MI family, T-Score, Z-Score).
  - Set the effect threshold (for example, `All values`, `top 10`, `top 50`, ...).
  - Appropriate effect-size measures are still being debated in the field, so the default setting is to show all values for this measure.
  - With default settings, keywords are ranked according to likelihood-measure scores (effectively ranking by p-values). This raises known methodological questions, but it remains the current standard in the field.
  - In practice, keyword rankings based on likelihood often produce more intrinsically intuitive results than rankings based directly on an effect-size measure.
- AI Settings
  - Edit the system prompt used when "AI Search" is enabled. This will guide the LLM during an interaction (see the KWIC tool settings for details).

Wordcloud Tool



The Wordcloud tool turns text or table outputs into a weighted word cloud. You can generate clouds directly

from a Scratchpad text area or from the exported results of other tools (KWIC, File View, Cluster, N-Gram, Collocate, Word, and Keyword). In the cloud, each item's size reflects its selected value (for example, frequency), and appearance options let you control layout, color, masking, and font scaling.

Practical Examples

- Research: Generate a cloud from a Keyword table to quickly inspect which key items dominate a target corpus before deeper statistical interpretation.
- Teaching: Build a cloud from Scratchpad text (for example, a lesson passage) to highlight repeated vocabulary for discussion and follow-up activities.
- Student use: Run a cloud from Word or N-Gram outputs to identify high-salience vocabulary patterns when preparing reports or presentations.

Core Workflow

1. Select a `Source`.
    - Choose `Scratchpad` to use free text, or choose another tool whose results are already available.
2. Set `Output label` and `Output value`.
    - For `Scratchpad`, `File View`, and `KWIC`, these default to `Word` and `Freq`.
    - For table-based sources, choose which column becomes the word label and which numeric column controls size.
3. Set `Max. Words` to control how many items are drawn.
4. (Optional) If using `Scratchpad`, enter or paste text in the Scratchpad panel.
5. Click `Start` to generate the cloud.
6. Review the visualization in the viewer pane.

Additional Features

- Scratchpad AI prompt: Enable `AI Prompt`, enter a prompt, and press Return to let AI generate Scratchpad text before creating the cloud.
- Masked clouds: Enable `Use mask` and choose a mask file (`.svg`/`.png`) to constrain cloud shape; you can also add custom masks.
- Mask styling: Use transparent-mask mode, show mask outline, and adjust outline width/color.
- Color modes: Choose `Color theme`, fixed `Text color`, or `Mask color (if available)`, then set range/background options.
- Font controls: Set font family, min/max font size, scaling factor (power transform), and optional log scaling.
- Clipboard/export: Copy the generated image to the clipboard or save it as a `.png` file.
- Validation checks: The tool warns if a previous generation is still running, if AI generation is already running, if no AI prompt is entered, or if the selected source has no exportable data.

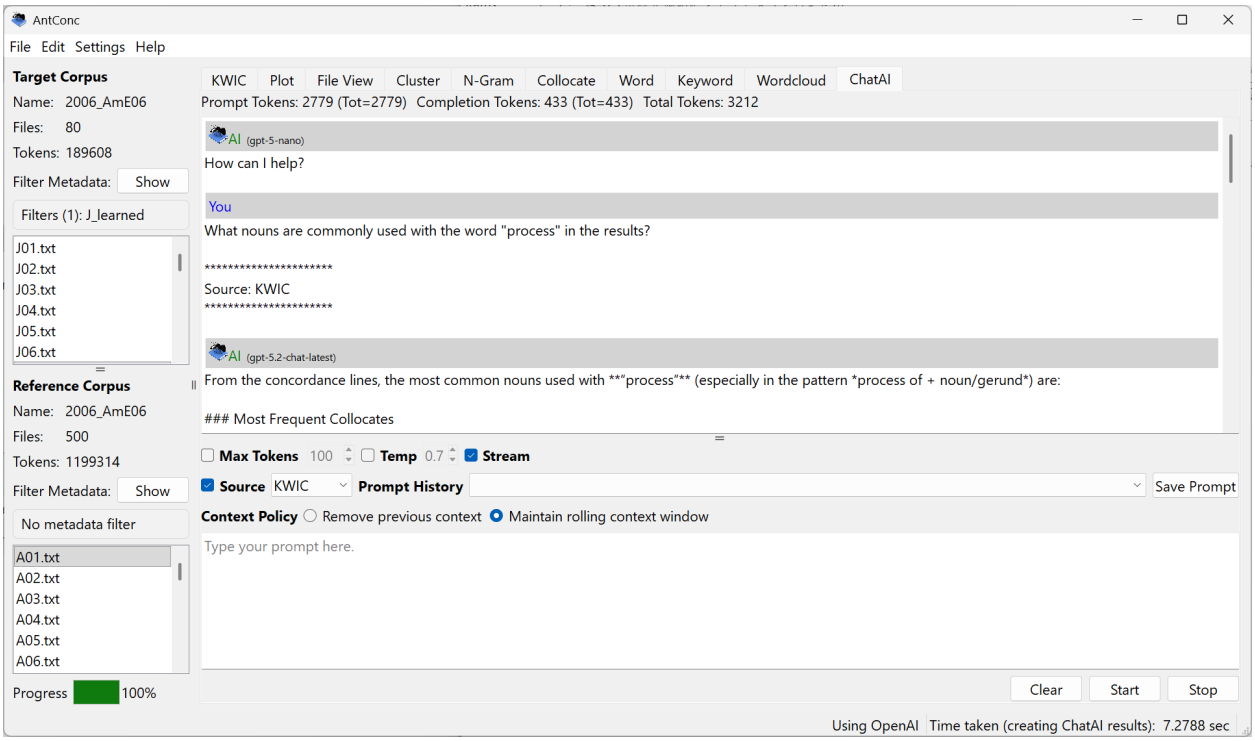
Tool Settings

Open via Settings -> Tool Settings -> Wordcloud Settings.

- General View

  - Margin: Controls spacing around placed words.
  - Canvas Grow Speed: Controls how aggressively the drawing canvas expands while fitting words.
  - Layout: Selects the placement method (`Distance Transform`, `Spiral`, or `Square`).
  - `Spiral` (default: Quickly packs words from a center point outwards).
  - `Distance Transform`: Slower, but better able to fit words into complex mask shapes.
  - `Square`: Also slow, but optimized for square, densely packed word clouds.
- Color Theme Options
  - Perceptually uniform sequential
  - These themes have incremental changes in lightness (and often saturation) that are perceived to be uniform. This makes them suitable for representing changes in frequency or other values.
  - Sequential
  - These themes have incremental changes in lightness (and often saturation). This makes them suitable for representing changes in frequency or other values.
  - Qualitative
  - These themes use miscellaneous colors. This usually makes them unsuitable for value-mapped interpretation, except when the goal is visual appeal rather than value-to-color correspondence.
- Mask Theme Options
  - Default / Regular / Brands / Solid shapes: Controls which mask categories are shown in the mask list.
- AI Settings
  - Edit the system prompt used when "AI Search" is enabled. This will guide the LLM during an interaction (see the KWIC tool settings for details).

ChatAI Tool



The ChatAI tool lets you interact with an AI model inside AntConc and optionally ground prompts in outputs from other tools. You can run direct chat prompts, append source data from a selected tool, manage conversation context, and monitor token usage while responses are generated.

Practical Examples

- Research: Attach KWIC or Cluster output as source data and ask the model to summarize recurring patterns before manual close reading.
- Teaching: Paste a prompt rubric, attach Word or Keyword results, and generate draft explanations or classroom discussion questions.
- Student use: Ask the model to explain frequency or collocation results in plain language, then refine with follow-up prompts in the same chat.

Core Workflow

1. Configure AI service/model in `Settings -> Global Settings -> AI` (local or cloud), including API keys where required.
2. Open ChatAI and set run options as needed:
    - `Max Tokens` to cap output length.
    - `Temp` to control response randomness.
    - `Stream` to show output incrementally.
3. Choose a context policy:
    - `Remove previous context` to treat each submission as a new interaction.
    - `Maintain rolling context window` to keep prior turns within the model context budget.
4. (Optional) Enable `Source` and select a tool (for example, KWIC, File View, Cluster, N-Gram, Collocate, Word, Keyword) to append the source results to your prompt to enable source-grounded responses.
5. Enter your prompt in the text box.
6. Click `Start` (or press `Shift+Enter`) to send.
7. Read the response in the viewer and continue the conversation, or click `Clear` to reset context.

Additional Features

- Prompt History: Save prompts with `Save Prompt` and reuse them from the `Prompt History` list.
- Stop generation: Use `Stop` to interrupt an active response.
- Token monitoring: The viewer shows prompt, completion, and total token counts for each interaction.
- Context overflow handling: When enabled in settings, ChatAI can show approximate context-length warnings and offer options such as sending anyway, shortening, or canceling.
- Compatibility fallbacks: If a model rejects explicit max-token or temperature parameters, ChatAI prompts you to retry with those options disabled.
- Service safety checks: The tool warns when required model/API configuration is missing.

Tool Settings

Open via `Settings -> Tool Settings -> Chat AI Settings`.

- Prompt Settings
    - System Prompt: Set the default system prompt prepended to ChatAI interactions.
- Warning Settings
    - Show (approximate) context length warnings: Toggle warnings when prompts are estimated to exceed available context.

Search query options

This section introduces practical ways to search your corpus using different strategies. You can run simple word or phrase lookups, widen matches with wildcards, use token-level regular expressions for precise patterns, or switch to full-text regex for advanced multi-word matching. Choosing the right search strategy helps you move from quick exploration to detailed pattern analysis.

Search strategies

- Simple searches
  - Use this strategy for straightforward word or phrase lookup.
  - Useful when you want exact lexical items without regex syntax.
  - Example 1 (single word): `language`
  - Example 2 (phrase across tokens): `corpus analysis`
- Wildcard searches
  - Use wildcard syntax for fast, flexible token-level matching when you do not need full regex patterns.
  - Useful for prefix/suffix families, alternatives, and quick pattern expansion.
  - Example 1 (starts with sequence): `teach*`
  - Example 2 (ends with sequence): `*ing`
  - Example 3 (alternatives): `student|pupil` or `[student,pupil]`
  - Example 4 (a word followed by a non-word marker): `Can I help_[?]` ("Can I help?")
  - The full list of available wildcards can be found in `Global Settings -> Searches`.
- Regex (token level) searches
  - Enable `Regex` to interpret each space-separated query token as a regex pattern.
  - Useful for precise token-level constraints while still searching token-by-token.
  - Example 1 (single-token regex): `colou?r`
  - Example 2 (two-token regex): `data.* driven`
- Full-text searches
  - Enable `Full Regex` (if available) to run one regex against full document text (not only token-by-token matching).
  - Useful for multi-word spans, cross-token patterns, and advanced constructs such as backreferences.
  - Example 1 (multi-word sequence with flexible spacing): `\bcorpus\s+linguistics\b`
  - Example 2 (backreference): `\b(\w+)\s+\1\b`

Notes:

- `Full Regex` is available only for raw-files corpora built with Full Regex storage enabled.
- When `Full Regex` is active, `Regex` is forced on, while `Words` and `Adv Search` are disabled for that run.
- Performance note: simple searches are optimized for speed. Wildcard searches typically have a small performance cost. Token-level regex searches usually have a similar but slightly higher cost. Full Regex cannot use the same fast database indexing path, so it has the largest performance cost and should be used with care on very large corpora (for example, many millions of words).

Advanced Searches

Advanced Search lets you build more controlled queries than the single main search box. It is useful when you want to search with long lists, add context constraints, or apply metadata-aware filters.

Feature availability by tool:

- Search Query List: KWIC, Plot, File View, Cluster, N-Gram, Collocate, Word, Keyword.

- Context Search: KWIC, Plot, File View, Cluster, Collocate, Word, Keyword.
- SQL Search: KWIC, Plot, File View, Cluster, Collocate, Word, Keyword.
- N-Gram uses a simplified Advanced Search dialog that supports Search Query List only.

Search Query List

Use this option to add many alternative queries without typing a long expression into the normal search box.

- Why this is useful: It makes repeated or large-set searching easier to manage and reuse.
- Typical use case: You have a long list of target words (for example, 50 reporting verbs) and want to run them together.
- How to add items:
  - Type a query and click `Add`.
  - Paste multiple lines (each line becomes one query item).
  - Drag and drop text lists into the query list box.

Context Search

Use this option to require (or exclude) context terms near your main search term, similar to collocational filtering.

- Why this is useful: It lets you focus on context-dependent meanings and co-occurrence patterns instead of raw keyword matching.
- Typical use case: Search for `university` only when `student` or `students` appears within a chosen left/right window.
- Important constraint: each context term must be a single item (one token), not a multi-word phrase.
- Main controls:
  - `Context Query` options (`Words`, `Case`, `Regex`).
  - `Search Logic` (`OR` or `AND`) across context terms.
  - `Window Span` (`From ...L` and `To ...R`).
  - `Filters` (`In context` / `Not in context`).

SQL Search

Use this option to apply advanced filtering conditions through custom SQL-style metadata joins.

- Why this is useful: It supports complex filtering based on custom metadata tables created when building a corpus, including metadata that is not automatically exposed through the main `Filter Metadata` control.
- Typical use case: Restrict results to files marked as `genre = 'academic'` in a custom metadata table.
- Input format:
  - Each entry should be a JSON array with three fields: `["table", "condition", "join_column"]`.
  - Example: `["genres", "genre = 'academic'", "doc_id"]`.

Menu Options

Menu options are divided into four groups: "File", "Edit", "Settings", and "Help". The options available in each group will be described below.

File

- Open File(s) as 'Quick Corpus'...
  - Opens a file picker so you can build and load a temporary "Quick Corpus" from selected files using the raw-files builder.
- Open Corpus Manager...
  - Opens the Corpus Manager dialog, where you can load target/reference corpora, build corpora, and manage corpus entries.
- Open AI Model Manager...
  - Opens the AI Model Manager dialog for selecting/managing local AI models used by AI features.
- Swap Target/Reference Corpora
  - Swaps the current target and reference corpus assignments.
- Clear Tool
  - Clears results in the currently active tool tab.
- Clear All Tools
  - Clears results in all tool tabs, but keeps currently loaded corpora.
- Clear All Tools and Files
  - Clears all tool results and clears loaded corpora/files from the corpus viewers.
- Save Current Tab Results...
  - Exports the currently active tool's result output (for example, text/TSV output or image output, depending on the tool).
- Save Current Tab Database Tables...
  - Exports the active tool's underlying database result tables as a ZIP package (when supported by that tool).
- Import Settings From File... / Export Settings To File...
  - Imports saved `.ini` settings into the interface, or exports current settings to an `.ini` file.
- Restore Default Settings
  - Reloads the default settings profile.
- Exit
  - Closes AntConc. Depending on Restore Settings preferences, current settings may be saved for the next session.

Edit

- Select All
  - Selects all visible results/content in the currently active tool view.
- Copy
  - Copies the current selection (or current tool copyable content, depending on the tool) to the clipboard.

Settings

Global Settings (applied to all tools in the interface)

- AI
  - Set the `Current service` used by AI features and optionally show/hide cloud-service warnings.
  - Configure cloud services (`OpenAI`, `Google Gemini`, `Claude`) with API keys, model selection, and `Update` model-list actions.

  - Configure local services (`AntConc (Llamafile)`, `LM Studio (Bridge)`) by setting endpoint/port values.
  - Choose whether model lists show `Recommended` or `All` models.
- Colors
  - Set `Search hit color` used in main search highlighting.
  - Set `Action widget color` used in manager interfaces (for example, corpus/AI model manager action widgets).
- Files
  - Set `Path View` (`Full path` or `Name only`) for file display.
  - Choose default/drag-drop corpus file extensions in `Corpus File Types (Default/Drag-Drop)`.
  - Set encoding error handling for corpus loading (`Show errors` or `Ignore errors`).
- Fonts
  - Set interface font family, size, and bold style, with a live sample preview.
- Language Direction
  - Choose `Left-to-right` or `Right-to-left` display behavior.
  - Enable `Arabic` when working with Arabic display/processing workflows.
- Restore Settings
  - Set whether the previous session settings should be restored automatically (`Yes`/`No`).
- Searches
  - Displays the wildcard and related search symbols used by the query system (for reference).
- Statistics
  - Set the `Normalized frequency multiplier` when displaying normalized frequencies.
  - Set `Floating-point precision` for displayed statistical values.
- Stoplist/Allowlist
  - Manage stoplist and allowlist word sets (including add/clear and paste/drop workflows).
  - Toggle `Hide punctuation` and enable/disable applying Stoplist/Allowlist filtering.
  - These filters are used in Cluster, N-Gram, Collocate, Word, Keyword, and Wordcloud.
- Tags
  - Display Type decides which word-level information is shown (for example, `Type`, `POS`, `Headword`, and combinations).
  - Important: available display types depend on what annotation is present in the loaded corpus. If a corpus does not contain POS or headword annotation, those related options will not be available.
  - `Type`: combines words with the same type into one entry, even if POS/headword tags differ; frequencies and range values are summed.
  - `Type+POS`: treats same-type words with different POS tags as separate entries; frequencies and range values are independent.
  - `Type+[POS]`: combines same-type words into one entry and also shows POS variants (with separate counts) that make up the total.
  - `Type+Headword`: treats same-type words with different headword (lemma) tags as separate entries; frequencies and range values are independent.
  - `Headword`: combines words from the same lemma family into one headword entry; frequencies and range values are summed.
  - `Headword+[Type]`: combines words from the same lemma family into one entry and also shows type variants (with separate counts) that make up the total.
  - Use `Set for all tools` to push the selected display type across tool interfaces.

Tool Settings

Tool-specific settings are documented in each tool's section above (KWIC, Plot, File View, Cluster, N-Gram, Collocate, Word, Keyword, Wordcloud, and ChatAI). Refer to those sections for current options and behavior.

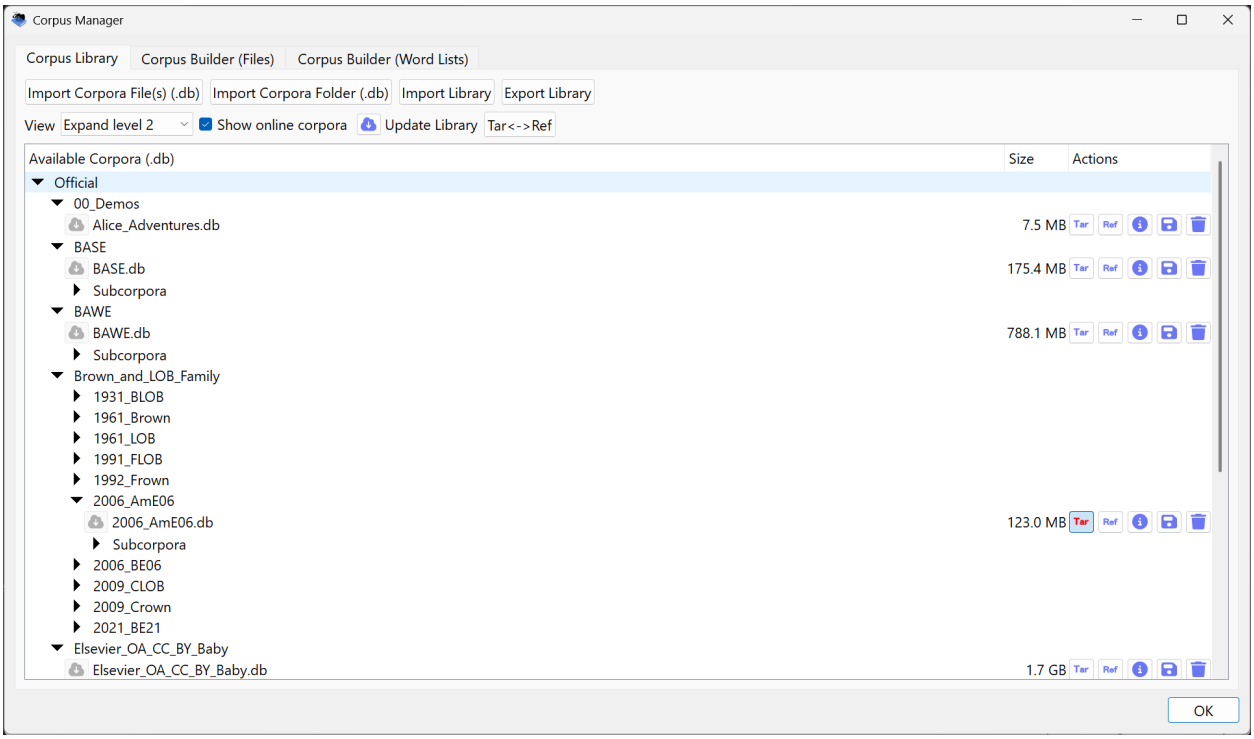
Corpus Manager

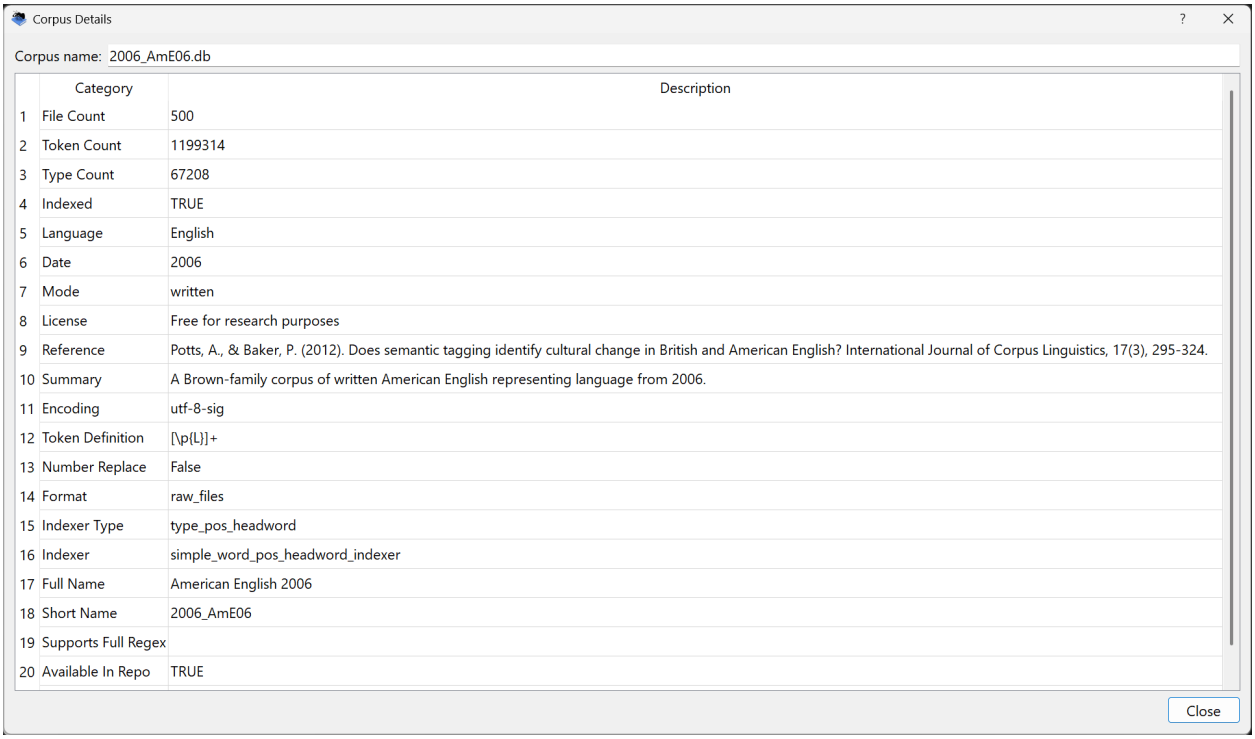
The Corpus Manager is where you load, organize, and build corpora for AntConc. It provides three tabs that serve different goals:

- `Corpus Library`: manage existing corpus databases (default online corpora and your user corpora), assign them as Target/Reference, and import/export library content.
- `Corpus Builder (Files)`: build a new corpus database from raw files (text, table, HTML/XML, Word, PDF, EPUB), with control over indexing, token definition, annotation/grouping inputs, metadata tables, and processing options.
- `Corpus Builder (Word Lists)`: build a corpus database from simple or advanced word-list tables (primarily for frequency/keyness workflows).

Compared with `Open File(s) as 'Quick Corpus'...`, the Corpus Manager gives you full corpus-construction control: you can assign a persistent corpus name, configure annotation-aware indexing, add metadata tables, customize token definitions and text-processing behavior, and save/reuse the resulting corpus in your library.

Corpus Library





	Category	Description
1	File Count	500
2	Token Count	1199314
3	Type Count	67208
4	Indexed	TRUE
5	Language	English
6	Date	2006
7	Mode	written
8	License	Free for research purposes
9	Reference	Potts, A., & Baker, P. (2012). Does semantic tagging identify cultural change in British and American English? International Journal of Corpus Linguistics, 17(3), 295-324.
10	Summary	A Brown-family corpus of written American English representing language from 2006.
11	Encoding	utf-8-sig
12	Token Definition	[\p{L}]+
13	Number Replace	False
14	Format	raw_files
15	Indexer Type	type_pos_headword
16	Indexer	simple_word_pos_headword_indexer
17	Full Name	American English 2006
18	Short Name	2006_AmE06
19	Supports Full Regex	
20	Available In Repo	TRUE

Corpus Library is used to manage and activate existing corpus databases (`.db`). It shows available default/online corpora and your user corpora, and provides action buttons to assign Target/Reference roles, inspect corpus info, save copies, and delete local entries.

Practical Examples

- Research: Keep a stable set of project corpora and quickly switch which corpus is Target vs Reference for keyword comparison.
- Teaching: Prepare class corpora in advance, then assign one as Target and one as Reference during live demonstrations.
- Student learning: Import a teacher-shared corpus database and immediately use it without rebuilding from raw files.

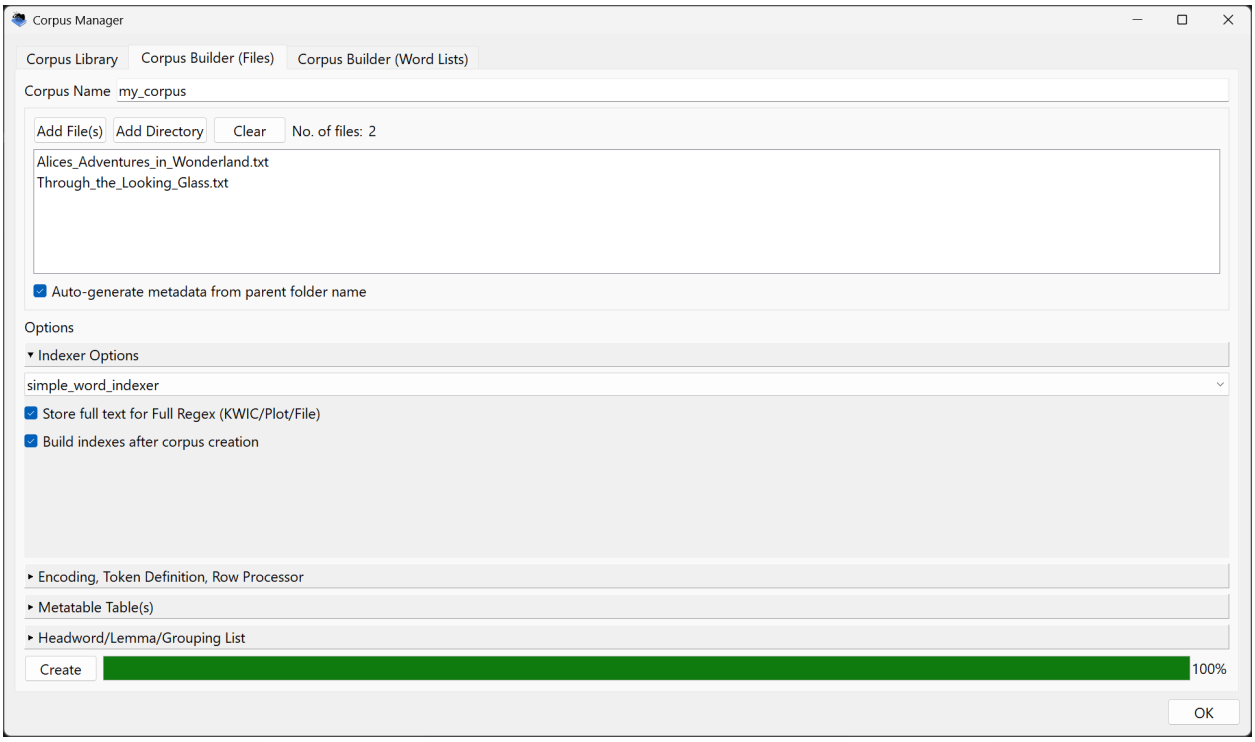
Core Workflow

1. Open `File -> Open Corpus Manager...` and select the `Corpus Library` tab.
2. Locate a corpus in the tree (`Default` and/or `User` groups).
3. If needed, download an online corpus (by clicking on the `Download from cloud` action button), or import a local `.db` corpus file/folder.
4. Use the action buttons for the selected corpus:
    - `Tar` to set as Target corpus.
    - `Ref` to set as Reference corpus.
    - `Info` to view corpus details.
    - `Save` to save the corpus to a new location.
    - `Delete` to remove the corpus from the library.
5. Close the Corpus Manager and run tools with the selected corpus roles.

Additional Features

- Corpus import options: Import individual `.db` files or recursively import `.db` files from a folder.
- Library import/export: Back up or restore the entire corpus library as a `.zip` archive.
- View controls: Expand/collapse tree levels to simplify the display.
- Online corpora: Toggle the display to `Show online corpora`.
- Library refresh: Click `Update Library` to reflect the latest online corpora.
- Role swap: Click the `Tar<->Ref` button in the library view to change the roles of the selected corpora.

Corpus Builder (Files)



Corpus Builder (Files) creates a reusable corpus database from source files. Unlike Quick Corpus, this builder lets you give the corpus a stable name and configure key processing decisions (indexer type, encoding, token definition, row processing, optional metadata tables, and optional headword/lemma/grouping mappings).

Practical Examples

- Research: Build a named project corpus with custom token settings and metadata tables so searches can be filtered by document variables.
- Teaching: Build a class corpus once, save it in the library, and reuse the same prepared dataset across lessons.
- Student learning: Create a personal corpus from assignment texts and tune token/annotation settings to match the analysis goal.

Core Workflow

1. Choose a name for your custom corpus.

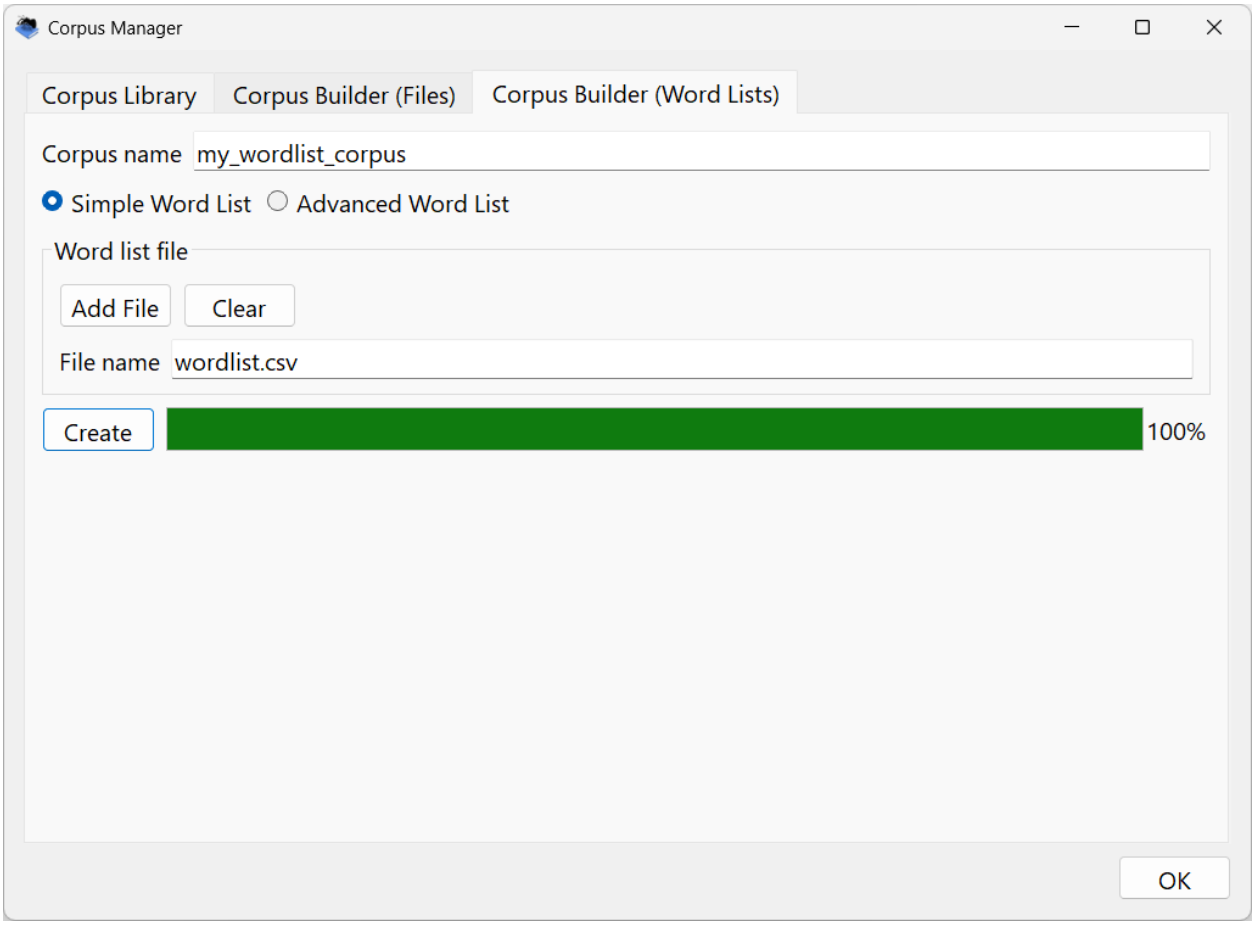
    - A default name is provided.
2. Choose the files to include.
    - Use `Add File(s)` or `Add Directory`.
    - Supported file types include plain text (`.txt`, `.srt`, `.sub`), table files (`.csv`, `.tsv`), HTML/XHTML/XML (`.html`, `.xhtml`, `.xml`), Word (`.doc`, `.docx`), PDF (`.pdf`), and EPUB (`.epub`).
3. Review the option settings (defaults should be suitable for basic corpora)
    - `Indexer`, `File Encoding`, `Token Definition`, `Metadata tables`, `Headword/Lemma/Grouping list`. For more details, see the Additional Features section.
4. Click `Create`. Built corpora are added to the `Corpus Library` and can be reused, exported, and reassigned as Target/Reference later.
5. In the completion dialog, optionally assign the new corpus immediately as `Target` or `Reference`.

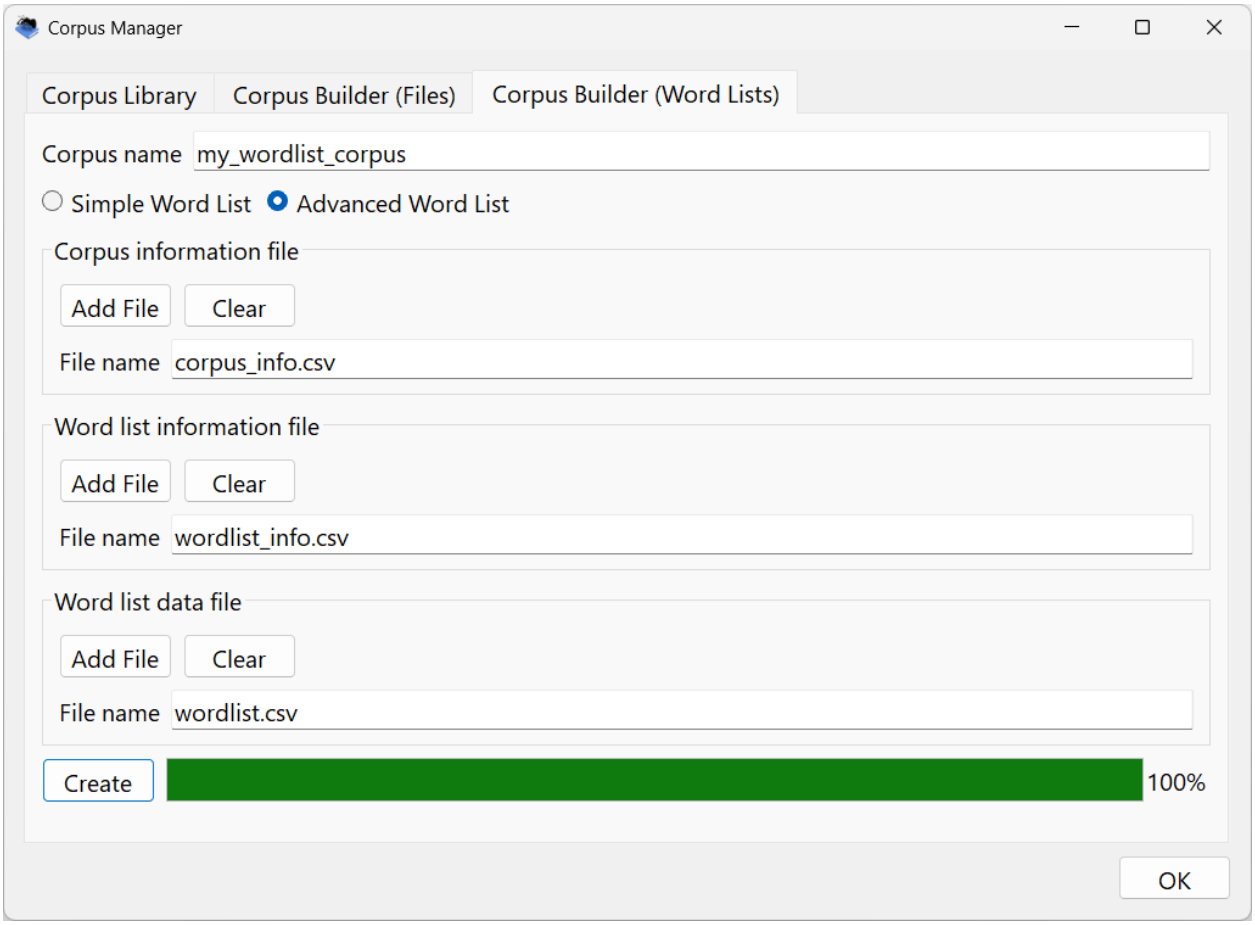
Additional Features

- Indexer options
    - Decide the indexer used to process files:
    - `simple_word_indexer`: suitable for files with no POS/headword annotation.
    - `simple_word_pos_headword_indexer`: suitable for files with POS/headword annotation.
    - `simple_word_bibertag_indexer`: suitable for files tagged in Biber-style format.
    - Additional indexers may become available for other file structures.
    - Full Regex preparation at build time: `Store full text for Full Regex (KWIC/Plot/File)` can be enabled for raw-files corpora.
- File encodings
    - UTF-8 is the default and usually the best first choice.
    - If you are unsure, start with UTF-8. If you later see encoding errors or corrupted display, switch to another encoding option.
    - For Word (`.docx`) and PDF (`.pdf`) files, the default encoding is usually suitable.
    - Further Unicode references: http://www.cs.tut.fi/~jkorpela/unicode/guide.html, http://www.unicode.org/, http://www.unicode.org/Public/5.0.0/ucd/UCD.html, http://www.unicode.org/Public/UNIDATA/PropList.txt, http://www.unicode.org/charts/.
- Token definition
    - Token definition determines what counts as a token/word in the corpus. The default is set as 'letters' of all the world's languages.
    - `Show Token Definition Settings` opens three modes:
        * Character Classes (Unicode-aware default)
        * User-Defined Characters
        * User-Defined Regex
    - Token settings also include:
        * `Ignore header`
        * `Ignore footer`
        * `Ignore non-embedded tags` (Any text between these two tags will be ignored when the corpus is created)
        * `Ignore embedded tags` (Any text appearing after the tag marker will be ignored when the corpus is created)
        * `Token Testing Area` (type/paste text, click `Test`, and preview generated tokens)
    - After defining token settings, click `Apply` in the Token Definition Settings dialog.
    - Choose row processing mode:
        * `One text per file` (default): each file becomes one corpus text.
        * `One text per row`: each row becomes a separate corpus text (useful for tabular datasets).
- Metadata tables (if available):
    - Add metadata table files (CSV/TSV) to store with the corpus database.

  - Metadata columns should align with corpus join keys (for example, `doc_id`).
  - Once built, these can be used as conditional filters in advanced SQL-style searches.
  - To understand the default table structure, open the corpus database in an SQLite database reader (e.g., https://sqlitebrowser.org/) and view the different tables.
- Headword/Lemma/Grouping list (if available):
  - Add an optional grouping list file to map words to headwords/lemmas/groups.
  - Existing headword/lemma/grouping values will be overwritten by this mapping.
  - Typical tab-separated format:
  - `headword<TAB>family member 1<TAB>family member 2...`
  - Once built, these mappings can be used in later conditional searches.

Corpus Builder (Word Lists)





Corpus Builder (Word Lists) creates a corpus database from word-list tables instead of raw text files. It is useful when you already have aggregated lexical statistics and want to reuse them in AntConc without re-building from source documents. Word-list corpora are primarily intended for keyword/reference workflows; using them in unsupported tools can produce errors.

Practical Examples

- Research: Import an existing frequency list from another corpus pipeline and use it as a structured reference corpus for keyword analysis.
- Teaching: Prepare a simplified course-specific lexical profile as a word-list corpus and compare student corpora against it.
- Student learning: Build a small personal reference corpus from exported word-list tables and use it to explore keyness differences.

Core Workflow

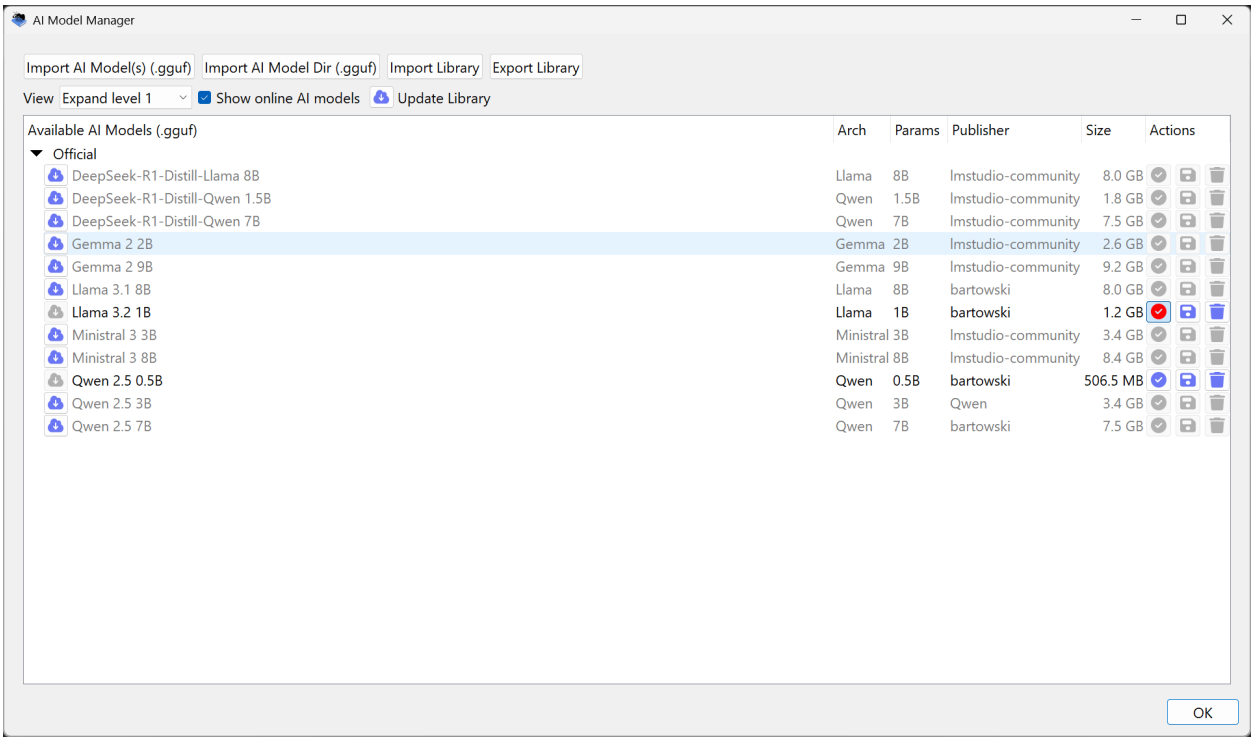
1. Open `Corpus Manager -> Corpus Builder (Word Lists)`.
2. Enter a `Corpus name`.
3. Choose the builder type:
    - `Simple Word List`
    - `Advanced Word List`

4. Load the required file(s) for the selected type.
    - For `Simple Word List`: add one file (CSV/TSV/TXT).
    - Expected columns: `type`, `freq`, and optional `range`.
    - For `Advanced Word List`: add three files (matching the format of saved word list tables in AntConc):
    - Corpus information file (CSV/TSV/TXT)
    - Word list information file (CSV/TSV/TXT)
    - Word list data file (CSV/TSV/TXT)
5. Click `Create`.
6. In the completion dialog, optionally assign the new corpus immediately as `Target` or `Reference`.

Additional Features

- Validation checks: the builder warns when required files are missing or no corpus name is entered.
- Overwrite confirmation: if a corpus with the same name already exists, you are prompted before replacement.
- Reusable output: built corpora are added to `Corpus Library` for later reuse.

AI Model Manager



AI Model Manager is used to manage local AI model files (`.gguf`) for AntConc. It mirrors the Corpus Library workflow, but for AI models instead of corpora: you can view available models, import your own models, select an active model, save copies, and manage your local model library. The local AI runtime is included with AntConc.

Practical Examples

- Research: Maintain a curated set of local models for private, reproducible analysis workflows.
- Teaching: Prepare one or more classroom-safe local models and switch quickly during demonstrations.
- Student learning: Import a provided model file and start using local AI features without cloud API setup.

Core Workflow

1. Open `File -> Open AI Model Manager...`.
2. Browse available models in the library tree (`Default` and/or `User` groups).
3. If needed, import model files:
    - `Import AI Model(s) (.gguf)` for individual files.
    - `Import AI Model Dir (.gguf)` for folder-based recursive import.
4. Use model actions for the selected entry:
    - `Select` to mark the model for local AI use.
    - `Save` to save/copy the model to another location.
    - `Delete` to remove the local model entry.
5. In `Settings -> Global Settings -> AI`, choose `AntConc (Llamafile)` as the current service to use selected local models in AI-enabled workflows.

Additional Features

- View controls: Expand/collapse tree levels and toggle `Show online AI models`.
- Library refresh: `Update Library` checks and refreshes available online model entries.
- Library import/export: Back up or restore the full AI model library as a `.zip` archive.
- Local+online organization: Keep default online entries and user-imported entries in one unified manager view.

Shortcuts

This section lists general purpose shortcuts used in input and table/list widgets that are usually available across multiple apps on your system, as well as custom shortcuts implemented specifically for AntConc.

Default Shortcuts

These are default shortcuts and work where the focused widget supports them.

- `Ctrl+C` (Windows) / `Command+C` (macOS): Copy selected text/items.
- `Ctrl+A` (Windows) / `Command+A` (macOS): Select all text/items.
- `Ctrl+V` (Windows) / `Command+V` (macOS): Paste clipboard text into editable fields/lists.
- `Ctrl+X` (Windows) / `Command+X` (macOS): Cut selected text in editable fields.
- `Ctrl+Z` (Windows) / `Command+Z` (macOS): Undo the last edit in editable fields.
- `Ctrl+Y` or `Ctrl+Shift+Z` (Windows) / `Command+Shift+Z` (macOS): Redo the last undone edit.
- `Delete` (Windows/macOS): Remove selected items in list widgets that support deletion.

- Arrow keys (Windows/macOS): Navigate values in combo boxes and spin boxes.
- `Ctrl+Tab` and `Ctrl+Shift+Tab` (Windows) / `Command+Option+Right` and `Command+Option+Left` (macOS, where available): Move between tabs in tabbed interfaces.

AntConc-Specific Shortcuts

- `Ctrl+1` to `Ctrl+0` (Windows) / `Command+1` to `Command+0` (macOS): Jump directly to a tool tab by index (1 to 10).
- `Ctrl+Shift+T` (Windows) / `Command+Shift+T` (macOS): Swap/toggle the target and reference corpora.
- `Ctrl+N` (Windows) / `Command+N` (macOS): Open the quick corpus creation dialog.
- `Ctrl+S` (Windows) / `Command+S` (macOS): Save the current results.
- `Ctrl+O` (Windows) / `Command+O` (macOS): Open the Corpus Manager.
- `Ctrl+Q` (Windows) / `Command+Q` (macOS): Quit AntConc.
- `Ctrl+G` (Windows) / `Command+G` (macOS): Open Global Settings.
- `Ctrl+T` (Windows) / `Command+T` (macOS): Open Tool Settings.
- `Ctrl+H` (Windows) / `Command+H` (macOS), KWIC tab only: Toggle the KWIC file-name column visibility and refresh results.
- `Ctrl+F` (Windows) / `Command+F` (macOS), File View: Move to the next hit in the current file hit list.
- `Ctrl+Shift+F` (Windows) / `Command+Shift+F` (macOS), File View: Move to the previous hit in the current file hit list.
- `+` (Windows/macOS), Plot tool controller focused: Zoom in the plot display.
- `-` (Windows/macOS), Plot tool controller focused: Zoom out the plot display.
- `=` (Windows/macOS), Plot tool controller focused: Reset plot zoom to 100%.
- `Ctrl+C` (Windows) / `Command+C` (macOS), Plot graphic focused: Copy the current plot image to the clipboard.
- `Shift+Enter` (Windows/macOS), ChatAI prompt box: Submit the prompt.
- `Delete` (Windows/macOS), KWIC results table: Remove selected rows from the current displayed results.
- `Shift+Delete` (Windows/macOS), KWIC results table: Leave only selected rows (exclude non-selected rows).
- `F1` (Windows) / `Command+?` (macOS): Open the About/Help action bound to the Help shortcut.

Citing/Referencing

For APA 7, software references should use an author-date-title-version-format-URL structure, without place of publication: Author. (Year). *Title* (Version) [Computer software]. Publisher. URL

Use this APA 7 format when you include AntConc in the reference list: Anthony, L. (YEAR OF RELEASE). *AntConc* (Version VERSION NUMBER) [Computer software]. Waseda University. https://www.laurenceanthony.net/software.ht

Example: Anthony, L. (2021). *AntConc* (Version 4.0.0) [Computer software]. Waseda University. https://www.laurenceanthony.net/software.html

For in-text citations, APA 7 allows two common approaches.

- Mention-only approach (for commonly used tools): Name the software and version in the text, with no reference-list entry.
  - `Analyses were conducted using AntConc (Version 4.0.0).`
- Cited approach (for less well known tools): Add the full reference-list entry and include an in-text author-year citation.
  - `Analyses were conducted using AntConc (Version 4.0.0; Anthony, 2021).`

For corpus-related publications, the `mention-only approach` is often suitable because AntConc is widely known in the field.

Statistics

Below is a list of statistics used in AntConc. The notation used here is taken from the work of Evert (2004: 36-37). Explanations are provided in Anthony (2023).

References:

Anthony, L. (2023). Common statistics used in corpus linguistics. Available at https://laurenceanthony.net/resources/statistics/

Evert, S. 2004. *The Statistics of Word Cooccurrences*: *Word Pairs and Collocations*. Unpublished Ph.D. thesis. University of Stuttgart. (Published 2005; available online at http://elib.unistuttgart.de/opus/volltexte/2005/2371/.)

Effect Size Statistics

$$\text{Dice coefficient} = \frac{2O_{11}}{R_1 + C_1}$$

$$\text{LogDice} = 14 + \log_2 \left(\frac{2O_{11}}{R_1 + C_1} \right)$$

$$\text{Log Ratio} = \log_2 \left(\frac{R_2 O_{11}}{R_1 O_{21}} \right)$$

$$\text{MI} = \log_2 \left(\frac{O_{11}}{E_{11}} \right)$$

$$\text{MI}^2 = \log_2 \left(\frac{(O_{11})^2}{E_{11}} \right)$$

$$\text{MI}^3 = \log_2 \left(\frac{(O_{11})^3}{E_{11}} \right)$$

$$\text{MS} = \min \left(\frac{O_{11}}{R_1}, \frac{O_{11}}{C_1} \right)$$

$$\text{Mu value} = \frac{O_{11}}{E_{11}}$$

$$\text{RRF} = \frac{R_2 O_{11}}{R_1 O_{21}}$$

$$\text{DRF} = \frac{O_{11}}{R_1} - \frac{O_{21}}{R_2}$$

$$z = \frac{O_{11} - E_{11}}{\sqrt{E_{11}}}$$

$$\text{T-score} = \frac{O_{11} - E_{11}}{\sqrt{O_{12}}}$$

For Log Ratio and RRF, if $O_{21} = 0$, a value of 0.5 is used.

Likelihood Statistics

$$\chi^2 \text{ (Chi Squared (4-term))} = \frac{(O_{11}-E_{11})^2}{E_{11}} + \frac{(O_{12}-E_{12})^2}{E_{12}} + \frac{(O_{21}-E_{21})^2}{E_{21}} + \frac{(O_{22}-E_{22})^2}{E_{22}}$$

$$\chi^2 \text{ (Chi Squared (4-term) - Yates Correction)} = \frac{(|O_{11}-E_{11}|-0.5)^2}{E_{11}} + \frac{(|O_{12}-E_{12}|-0.5)^2}{E_{12}} + \frac{(|O_{21}-E_{21}|-0.5)^2}{E_{21}} + \frac{(|O_{22}-E_{22}|-0.5)^2}{E_{22}}$$

$$\text{Log Likelihood (4-term)} = 2\left(O_{11} \ln\left(\frac{O_{11}}{E_{11}}\right) + O_{21} \ln\left(\frac{O_{21}}{E_{21}}\right) + O_{12} \ln\left(\frac{O_{12}}{E_{12}}\right) + O_{22} \ln\left(\frac{O_{22}}{E_{22}}\right)\right)$$

$$\text{Log Likelihood (2-term)} = 2\left(O_{11} \ln\left(\frac{O_{11}}{E_{11}}\right) + O_{21} \ln\left(\frac{O_{21}}{E_{21}}\right)\right)$$

$$\text{Text Dispersion (4-term)} = 2\left(O_{11} \ln\left(\frac{O_{11}}{E_{11}}\right) + O_{21} \ln\left(\frac{O_{21}}{E_{21}}\right) + O_{12} \ln\left(\frac{O_{12}}{E_{12}}\right) + O_{22} \ln\left(\frac{O_{22}}{E_{22}}\right)\right)$$

$$\text{Text Dispersion (2-term)} = 2\left(O_{11} \ln\left(\frac{O_{11}}{E_{11}}\right) + O_{21} \ln\left(\frac{O_{21}}{E_{21}}\right)\right)$$

For Text Dispersion, O and E are for range values.

Community and Support

- Join the AntConc Discussion Group for questions, bug reports, and feature suggestions: https://groups.google.com/g/antconc
- If AntConc is useful in your research, teaching, or learning, you can support development via Patreon, Buy Me a Coffee, or PayPal. Current support links are listed on the AntConc software page: https://www.laurenceanthony.net/software/antconc/
- The development of *AntConc* has been supported by a Japan Society for Promotion of Science (JSPS) Grant-in-Aid for Scientific Research (C): No. 23501115, a Japan Society for Promotion of Science (JSPS) Grant-in-Aid for Young Scientists (B): No. 18700658, a Japan Society for Promotion of Science (JSPS) Grant-in-Aid for Young Scientists (B): No. 16700573, and a WASEDA University Grant for Special Research Projects: No. 2004B-861.