

Getting Started with ShinyConc™

Quick & Dirty¹ Tutorial & Setup

ShinyConc Builder allows you to build a custom concordancer in only a matter of minutes. To show you how to set up your own concordancer, this tutorial will use the Inaugural Address Corpus (available at <https://archive.org/details/Inaugural-Address-Corpus-1789-2009>).

Once you are finished with setup, you will have a concordancer that looks like the following:

Inaugural Corpus

The screenshot shows a ShinyConc concordancer interface. On the left, there are three dropdown menus for 'Year' (1789, 1793, 1797, 1801), 'President' (Adams, Buchanan, Bush, Carter), and 'Party' (Democratic, Democratic-Rep, Federalist, Independent). Below these is a search box containing 'freedom' and a search mode section with options for 'string', 'word' (selected), 'regular expression', and 'case-sensitive'. The main area displays search results for 'freedom' in a table with columns for 'file', 'left', 'center', and 'right'. The results show the word 'freedom' highlighted in red in the center column across various files like '1801-Jefferson.txt' and '1825-Adams.txt'. At the bottom, there is a 'Download as csv' button and a pagination control showing 'Showing 1 to 10 of 183 entries'.

Figure 1: An example of a custom concordancer made using ShinyConc

If you encounter any difficulties, please feel free to contact Dr. Christoph Wolk (project head and programmer) at Christoph.B.Wolk@anglistik.uni-giessen.de, or Bridgit Fastrich (project coordinator) at Bridgit.C.Fastrich@anglistik.uni-giessen.de. We are happy to help and grateful if you report any bugs you encounter while using the builder!

This tutorial is divided into four main parts:

Part 1: Acquiring the necessary programs and documents before you begin (pg. 2)

Part 2: Creating your metadata file and preparing your data (pg. 2)

Part 3: Setting up your concordancer in the ShinyConc Builder (pg. 5)

Part 4: Running your concordancer (pg. 9)

¹ We plan to develop more detailed documentation in the future and are available for help if you have any questions.

Part 1: Acquiring the necessary programs and documents

In order to create your own ShinyConc concordancer, you need the following:

- **R** and **R Studio** running on your computer. If you do not have these yet, go to <https://cran.r-project.org/> to get R and <https://www.rstudio.com/products/RStudio/> to get R Studio (Open Source Edition).
- your **corpus**, either as a single .csv file or a folder containing plain text files. See Part 2 for more details about the structure of your data.
- a **metadata file** that must have a specific format. See Part 2 for more details

Part 2: Preparing your corpus and creating your metadata file

The ShinyConc builder can support corpora contained in either files (‘.txt’) or spreadsheets (‘.csv’).²

If your corpus is stored in files (‘.txt’)

If your corpus is stored as .txt files (as is the case with the Inaugural Corpus), the directory in which it is contained might look as follows:

Name	Änderungsdatum	Typ	Größe
1789-Washington	13.05.2017 20:47	Textdokument	9 KB
1793-Washington	13.05.2017 20:47	Textdokument	1 KB
1797-Adams	13.05.2017 20:47	Textdokument	14 KB
1801-Jefferson	13.05.2017 20:47	Textdokument	10 KB
1805-Jefferson	13.05.2017 20:47	Textdokument	13 KB
1809-Madison	13.05.2017 20:47	Textdokument	7 KB
1813-Madison	13.05.2017 20:47	Textdokument	7 KB
1817-Monroe	13.05.2017 20:47	Textdokument	20 KB
1821-Monroe	13.05.2017 20:47	Textdokument	26 KB
1825-Adams	13.05.2017 20:47	Textdokument	18 KB

Figure 2: A screenshot of a corpus saved as files

Creating your metadata file for a .txt corpus

1. One column in your .csv file **must** be called “file” (**important:** all lowercase) and contain the full file name of each file of the corpus relative to the main directory of the corpus. In the case of the

²If you have not yet constructed your corpora: Corpora stored in files usually contains whole texts in each file (e.g. a corpus of newspaper articles or Wikipedia entries), or one small unit per file (e.g. a corpus with sentence or turn/utterance-level annotation, e.g. television dialogues or email sentences annotated for speech acts). Corpora stored in spreadsheets, in contrast, might contain question and answer pairs or other smaller units of analysis.

Inaugural Corpus, this file extension is simply the name of the file, as there are no subdirectories for this corpus (e.g. a division between spoken and written, or German and English language, etc.)

file	Year	President	Party
1789-Washington.txt	1789	Washington	Independent
1793-Washington.txt	1793	Washington	Independent
1797-Adams.txt	1797	Adams	Federalist
1801-Jefferson.txt	1801	Jefferson	Democratic-Republican
1805-Jefferson.txt	1805	Jefferson	Democratic-Republican
1809-Madison.txt	1809	Madison	Democratic-Republican
1813-Madison.txt	1813	Madison	Democratic-Republican
1817-Monroe.txt	1817	Monroe	Democratic-Republican
1821-Monroe.txt	1821	Monroe	Democratic-Republican
1825-Adams.txt	1825	Adams	Democratic-Republican

Figure 3: Listing the file names in your metadata file

- The rest of the columns describe the meta for your corpus. This includes any annotations (e.g. groupings, subdivisions) that you have and might want to have displayed in your concordancer. Ideally, the labels should be named as you want them to be displayed in your concordancer. In the Inaugural Corpus, the relevant groupings pertain to the year in which the address was given, the President who gave it, and the political party to which this President belonged.

file	Year	President	Party
1789-Washington.txt	1789	Washington	Independent
1793-Washington.txt	1793	Washington	Independent
1797-Adams.txt	1797	Adams	Federalist
1801-Jefferson.txt	1801	Jefferson	Democratic-Republican
1805-Jefferson.txt	1805	Jefferson	Democratic-Republican
1809-Madison.txt	1809	Madison	Democratic-Republican
1813-Madison.txt	1813	Madison	Democratic-Republican
1817-Monroe.txt	1817	Monroe	Democratic-Republican
1821-Monroe.txt	1821	Monroe	Democratic-Republican
1825-Adams.txt	1825	Adams	Democratic-Republican

Figure 4: Examples of annotation to be included in your metadata file

If your corpus is stored in a .csv file (e.g. QA pairs)³

1. If your corpus contains sequential data (e.g. dialogue lines, or sentence-level annotation), this data should be contained in one column, with each new line of data on a row. This column should also be called “text” (**important:** all lowercase). The below screenshot shows what a .csv file containing the Inaugural Address corpus and all relevant metadata would look like:

text	Year	President	Party
Fellow Citizens of the Senate and the House of	1789	Washington	Independent
Fellow Citizens: I am again called upon by the voice of	1793	Washington	Independent
When it was first perceived, in early times, that no	1797	Adams	Federalist
Friends and Fellow-Citizens, Called upon to undertake	1801	Jefferson	Democratic-Republican
Proceeding, fellow citizens, to that qualification which	1805	Jefferson	Democratic-Republican
Unwilling to depart from examples of the most	1809	Madison	Democratic-Republican
About to add the solemnity of an oath to the	1813	Madison	Democratic-Republican
Fellow-Citizens: I shall not attempt to describe the	1817	Monroe	Democratic-Republican
Fellow-Citizens: I shall not attempt to describe the	1821	Monroe	Democratic-Republican
In compliance with an usage coeval with the existence of	1825	Adams	Democratic-Republican

Figure 5: Example of the Inaugural Corpus contained in a .csv file

As can be seen in the above, the rest of your annotation should be displayed in the remaining columns. Again, the labels should be named as you want them displayed in your concordancer.

2. If your data consists of QA pairs, then your questions and answers should be in two separate columns and called “Questions” and “Answers,” respectively (**important:** this time uppercase!). In the screen below, an example of such a corpus is given.

Questions	Answers
Leider kommt es immer wieder vor, dass ich aus	Hallo Christoph, die fertig zubereiteten Burger sollten
Wird der McRib in Deutschland genau so herge-	Hallo Leischna, wir können hier nur für McDonald's De
warum werden Burger verkauft, obwohl sie scho	Hallo nach Aachen, du hast Recht, unsere Produkte
Schmeckt das mc Donalds essen genau wie wie	Hallo Jan-Niklas, es gibt weltweit fünf Produkte die eg
Warum gibt es über Euren Hauptlieferanten OSI	Hallo Jens, nein, wir haben nichts zu verbergen. Auch
Warum sind die Pommes immer so schnell kalt?	Hallo Amanira, unsere Pommes werden sehr dünn ge
Warum schmeckt der Mocca Frappe nicht so lech	Liebe Regina, schade, dass du diesen Eindruck hast
Kann man süß saure soße einzeln in größerer S	Hallo nach Aachen, wir hoffen, du hast den Super Bc

Figure 6: Example of a Q/A corpus contained in a .csv file

Creating your metadata file for a .csv corpus

Your metadata can be included in the same .csv file in the adjacent columns. See Figure 4.

That’s all you need to do to prepare your data and metadata.

³If you are using spreadsheet software such as Microsoft Excel, your file will have to be converted to .csv. Only do this very carefully and make sure to keep a copy of your old version so that you do not lose any valuable data or annotations in the conversion process!

Part 3: Setting up your concordancer in the ShinyConc Builder

Now that your data and meta-data have been prepared for use in the ShinyConc Builder, it's time to create your concordancer. Just follow the steps below.

1. **Load metadata:** Go to <http://shinyconc.de/builder/> and click *Browse...* to load your metadata.⁴

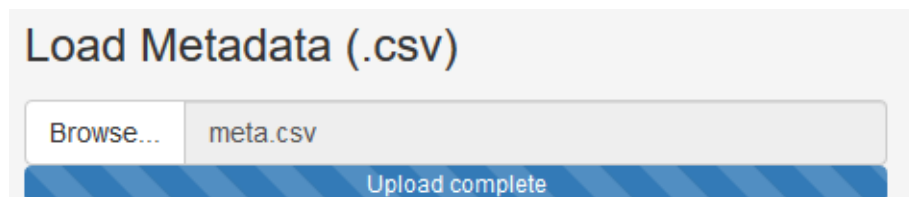


Figure 7: Loading your metadata

2. **App name:** Give your concordancer a name, which will be displayed at the top of the concordancer.

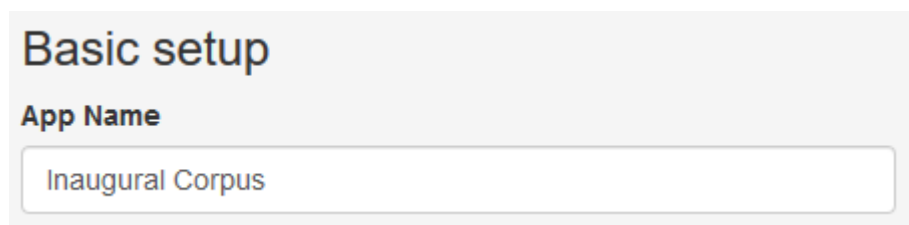


Figure 8: Naming your concordancer

3. **Corpus type/Read corpus from:** Indicate the type of corpus you have (a corpus of texts, lines, or Q/A pairs) and the type of data the corpus should be read from (i.e. the form in which your data is saved). In the case of the Inaugural Corpus, we have each address saved as individual text files, so we choose “texts” and “text file.”

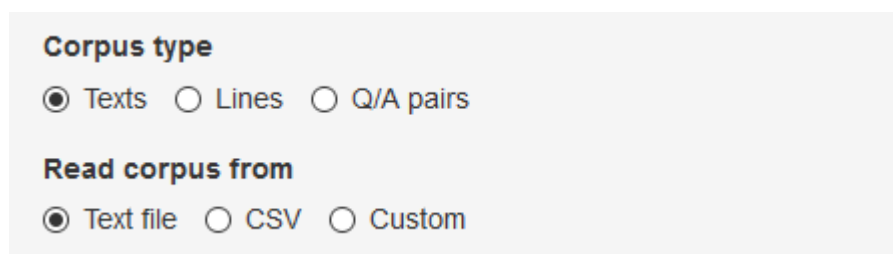


Figure 9: Selecting your corpus type and form from which the corpus should be read

4. **Character Encoding:** In some types of files, it might be useful to choose which type of character encoding is being used. If you are unfamiliar with this, keep the setting at “Auto.”

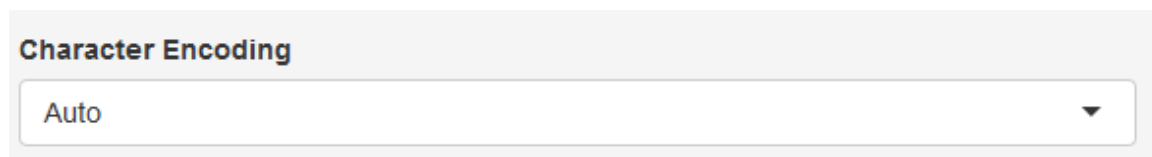


Figure 10: Selecting your character encoding

⁴In this panel, there are a few more options pertaining to how your metadata is formatted. Generally, you should be able to leave these at their default settings.

5. **Selector definitions:** Here you can configure your filtering tools for your concordancer. The selectors you will see displayed are automatically detected in your metadata and contain the headings of the columns in your data. **IMPORTANT:** In this step, you only have to configure the variables that you want display in your app. For the Inaugural Address concordancer, for example, we want the variables *Year*, *President* and *Party* to be displayed, but not the individual files, so we only have to configure the latter three variables.

! **Error note:** If your metadata in your selector definitions are not displayed individually, i.e.:

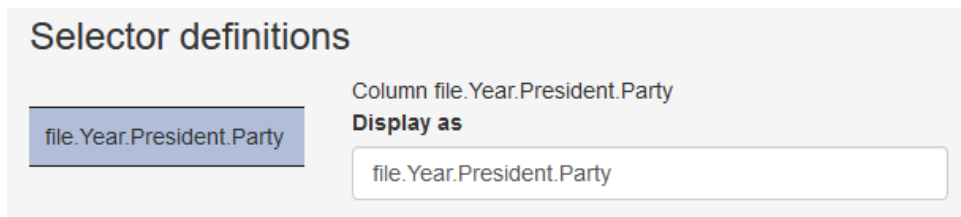


Figure 11: Erroneous display of variables due to faulty delimitation selection

This means that you have selected the wrong separation delimitation (see footnote 4). Try the other separator option (comma or semicolon) to see if that solves your problem.

Configuring the selectors: In the following, the configuration of the variables *Year* and *Party* will be used to help illustrate the various options.

5.1 **Display as:** This lets you choose what the selector will be called in the concordancer. The heading you used in your metadata will automatically appear here but you can change it here if preferred.

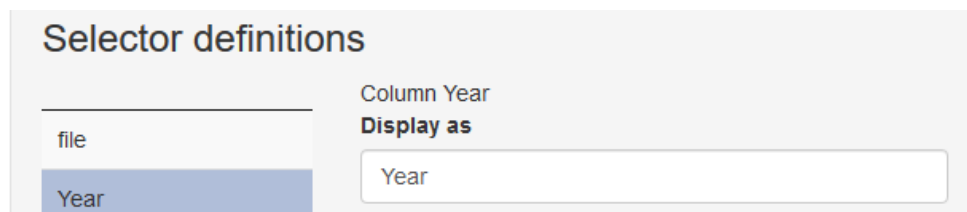


Figure 12: Labelling your variable filters

5.2 **Choose as:** This lets you choose the form in which your selector options will appear. You have the option between a **List**, **Radio Buttons**, or **Checklist**, which will be illustrated below.

List: A “list” is most appropriate when you have so many options to choose from that a scrolling function is preferred over displaying all options at once (cf. checklist), and when the user should be able to select more than one option. This is the case with the *Year* selector.

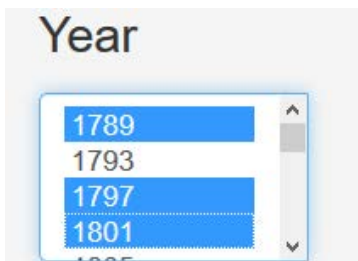


Figure 13: An example of the list display; here only search results from the years 1789, 1797 and 1801 would be shown.

Radio buttons: These are most appropriate when you want the user to be able to select *only one* of the predefined values. For example, if the user should only be able to choose one of the political parties, then these values would be displayed as follows and users could only select one party:

Party

- Democratic
- Democratic-Republican
- Federalist
- Independent
- Republican
- Taylor
- Whig

Figure 14: An example of radio buttons, in which only one value can be selected.

Checkboxes: Checkboxes are appropriate if the user should be able to select more than one of the options, and the list is short enough that it should be displayed in its entirety in the app. Here, we again take the example of Party, which is the only variable from the Inaugural Corpus that has few enough options to display them all at once. Other examples might include annotation categories such as speech act, sex of speaker, age ranges, etc.

Party

- Democratic
- Democratic-Republican
- Federalist
- Independent
- Republican
- Taylor
- Whig

Figure 15: An example of checkboxes, in which more than one value can be selected.

5.3 Choices: In a future version of ShinyConc Builder, this will allow you to choose the values that are displayed and how they are named (i.e. their labels), and will also provide a cascading option. This function is not yet active and should be left in its default setting.

5.4 Select by default: Here you get to choose which values/options are already preselected. For example, in the checkboxes selector for political party above, if you only want the values **Democratic** and **Republican** to be pre-selected, click **Choose:** and select those values

Select by default

All Choose:

Democratic Republican

Figure 16: Preselect only Democratic and Republican in the Party variable

6. Selector Order

This is where you choose how the variables included in your annotation will be displayed in the concordancer (the left-hand panel depicted in Figure 1 on page 1). In the case of the Inaugural Corpus, we have the selectors Year, President, and Party. Let's say that we want year to be displayed in its own row, and the President and Party on the next rows. Selectors that should be on the same row should be separated with a comma (','), and Selectors that should be on separate rows should be separated with a semicolon (';'), so we would have the following configuration:

Selector Order

Enter the names of the columns to show in the order they are supposed to appear. Separate Selectors on the same row with a comma (','), separate rows with a semicolon (;).

Selector Order

Year; President, Party

Figure 17: Determining the order in which your variables will be displayed in your concordancer.

This would yield the following result:

Year

- 1789
- 1793
- 1797
- 1801
- 1805

President

- Adams
- Buchanan
- Bush
- Carter

Party

- Democratic
- Democratic-Republican
- Federalist
- Independent
- Republican
- Taylor
- Whig

Figure 18: Example display in which Year is displayed on the first row and President and Party displayed on the second.

Note here that for this display, Year and President use the List and Party uses checkboxes, for which Democratic and Republican have been preselected (this would be the result of Figure 16).

If we wanted all three to be displayed on the same line, and to have all of them in the form of lists, the result we would get is also depicted in Figure 1.

7. Tool options

The last part of the concordancer setup concerns configuring the concordancer tools.

7.1 Extra columns in KWIC View

It can be helpful to display additional data in your KWIC view, for example the file name, an ID, the current speaker in a dialog corpus or specific variables of research interest. Simply type the name of the columns you want to show, and these columns will show up to the left of the KWIC view. In the example below, President and Party were specified to be shown left of the KWIC view.



Figure 19: Settings for KWIC view

President	Party	left	center	right
Jefferson	Democratic-Republican	the bar of the public reason;	freedom	of religion; freedom of the p
Jefferson	Democratic-Republican	reason; freedom of religion;	freedom	of the press, and freedom of
Jefferson	Democratic-Republican	on; freedom of the press, and	freedom	of person under the protectio
Jefferson	Democratic-Republican	rumental to the happiness and	freedom	of all. Relying, then, on the
Jefferson	Democratic-Republican	n institution so important to	freedom	and science are deeply to be
Jefferson	Democratic-Republican	airly and fully made, whether	freedom	of discussion, unaided by pow
Madison	Democratic-Republican	d personal rights, and of the	freedom	of the press; to observe econ
Adams	Democratic-Republican	y lot of humanity secured the	freedom	and happiness of this people.
Adams	Democratic-Republican	parable from the enjoyment of	freedom	, but which have more than onc
Adams	Democratic-Republican	buse of power consists in the	freedom	, the purity, and the frequenc

Figure 20: Resulting concordancer KWIC display with the search word freedom

7.2 Extra columns in Data View

Similarly, it can be useful to show more than just the actual text in the so-called data view. Just add the columns you want, as in the previous step. This field will be automatically populated with values depending on your corpus type; “text” for file and line corpora, “Question” and “Answer” for Q&A corpora. Do not delete them, or they will not show up in your concordancer.

Part 4: Running your concordancer

We’re almost there! Now that you’ve finished configuring your concordancer, it’s time to click **Create App** in the upper right panel:

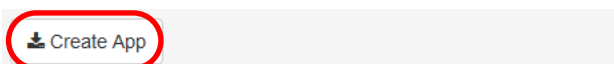


Figure 21: Create App button in the upper left panel of ShinyConc Builder

A window will then pop up asking you where you want to save your .zip file that contains all of the necessary documents. Now, just follow the remaining steps to get your concordancer up and running.

1. Unpack your zip folder. You will now have a folder called ShinyApp which contains the following files and subfolders:

corpus	21.05.2017 11:53	Dateiordner	
www	21.05.2017 11:53	Dateiordner	
	21.05.2017 11:53	RHISTORY-Datei	1 KB
app	21.05.2017 11:53	R-Datei	3 KB
ShinyApp	21.05.2017 11:53	R Project	1 KB
shinyconc.yaml	21.05.2017 11:53	YAML-Datei	1 KB

Figure 22: The unzipped folder with all the documents you need to run your concordancer. Just paste your corpus in the 'corpus' folder!

2. In the **corpus** folder, you now need to insert all of your corpus files. Note that the metadata file is already in the corpus folder and therefore does not need to be inserted again.
3. Open the R file called **app** in R Studio and click Run App (see below).

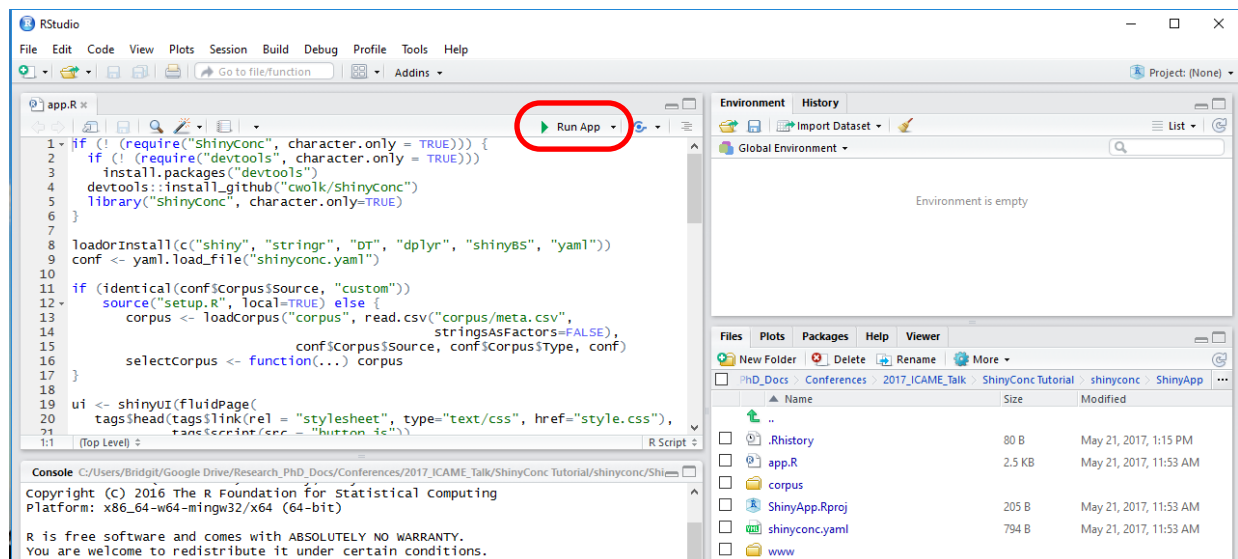


Figure 23: View of R Studio with the script loaded. Just press Run App!

A window in your preferred web browser should open automatically. Please wait a moment for the app to load (this will depend on the size of your particular corpus). In a couple of minutes, however, your concordancer should load and be ready to use.

Enjoy!