

# Corpus

Load a corpus of text documents, (optionally) tagged with categories, or change the data input signal to the corpus.

## Inputs

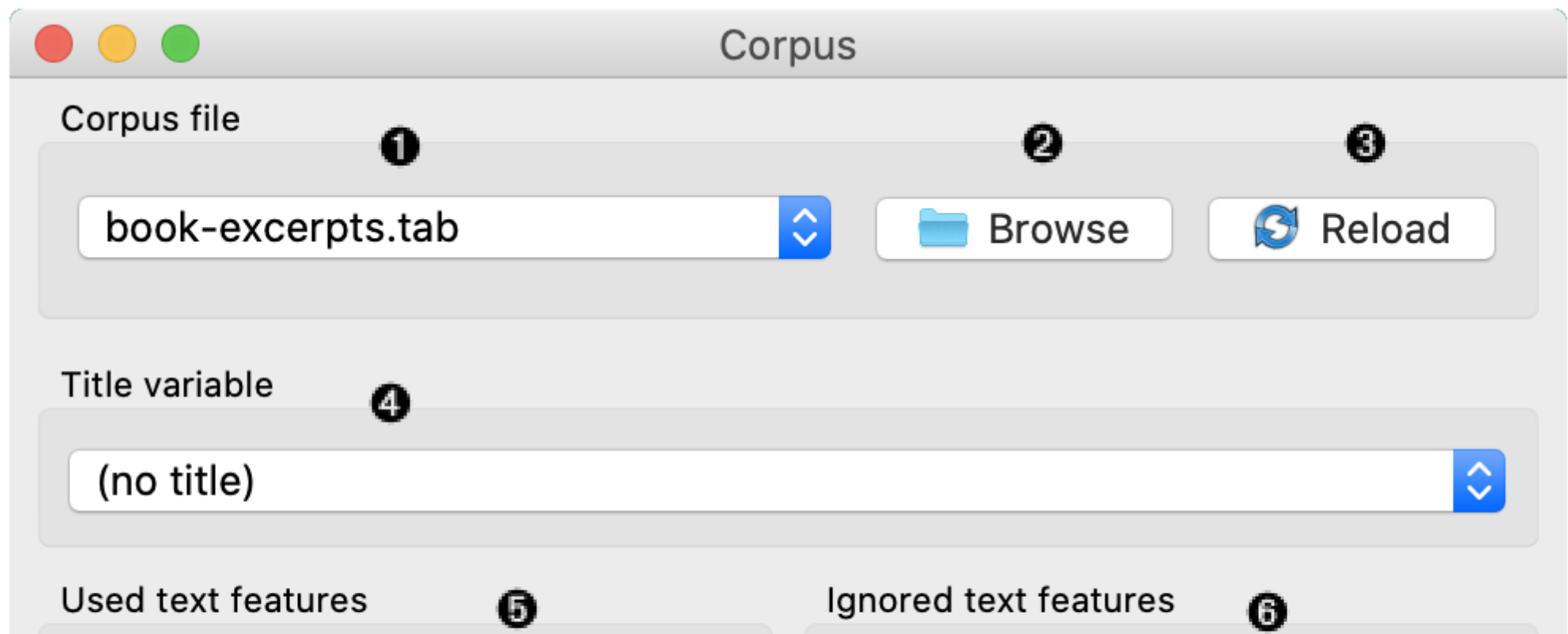
- Data: Input data (optional)

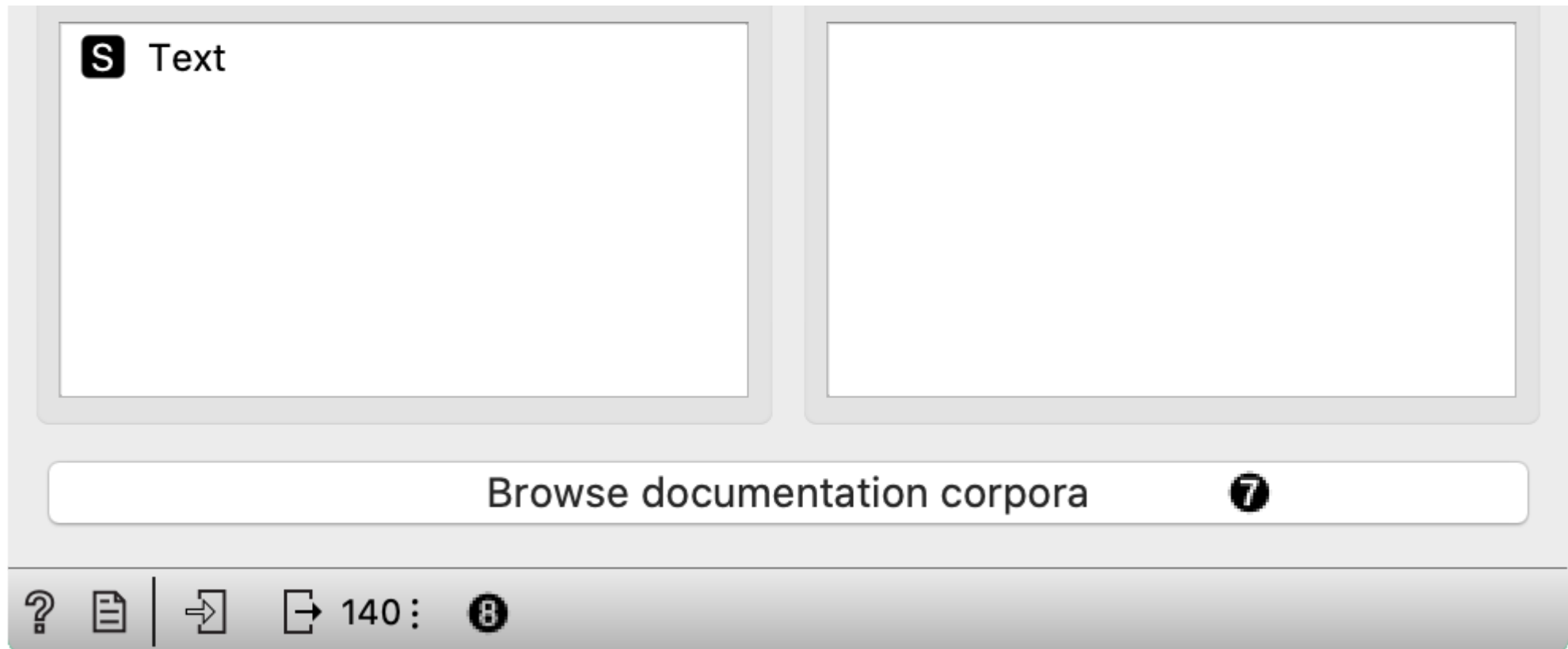
## Outputs

- Corpus: A collection of documents.

**Corpus** widget can work in two modes:

- When no data on input, it reads text corpora from files and sends a corpus instance to its output channel. History of the most recently opened files is maintained in the widget. The widget also includes a directory with sample corpora that come pre-installed with the add-on. The widget reads data from Excel (**.xlsx**), comma-separated (**.csv**) and native tab-delimited (**.tab**) files.
- When the user provides data to the input, it transforms data into the corpus. Users can select which features are used as text features.





1. Browse through previously opened data files, or load any of the sample ones.
2. Browse for a data file.
3. Reloads currently selected data file.
4. Select the variable that is shown as a document title in Corpus Viewer.
5. Features that will be used in text analysis.
6. Features that won't be used in text analysis.
7. Browse through the datasets that come together with an add-on.
8. Access help, make a report and get information on the loaded data set.

You can drag and drop features between the two boxes and also change the order in which they appear.

## Example

The first example shows a very simple use of **Corpus** widget. Place **Corpus** onto canvas and connect it to **Corpus Viewer**. We've used *book-excerpts.tab* data set, which comes with the add-on, and inspected it in **Corpus Viewer**.

The screenshot displays the Orange Data Mining software interface. At the top, a workflow diagram shows a 'Corpus' widget connected to a 'Corpus Viewer' widget. Below this, the 'Corpus' widget window is open, showing settings for the file 'book-excerpts.tab'. The 'Title variable' is set to '(no title)'. Under 'Used text features', 'Text' is selected. The 'Corpus Viewer' window is also open, displaying a list of 17 documents. The 'Info' tab shows document statistics: 140 documents, preprocessed (False), tokens (n/a), types (n/a), POS tagged (False), N-grams range (1-1), and matching (140/140). The 'Search features' and 'Display features' sections both show 'Category' and 'Text'. The 'RegExp Filter' is empty. The 'Text' tab displays the content of 'Document 1', which is a paragraph from a story about a child named Jim.

**Corpus Viewer Info:**

- Documents: 140
- Preprocessed: False
  - Tokens: n/a
  - Types: n/a
- POS tagged: False
- N-grams range: 1-1
- Matching: 140/140

**Search features:**

- Category
- Text

**Display features:**

- Category
- Text

☐ Show Tokens & Tags

☒ Auto send is on

**Document List:**

| Index | Document Name |
|-------|---------------|
| 1     | Document 1    |
| 2     | Document 2    |
| 3     | Document 3    |
| 4     | Document 4    |
| 5     | Document 5    |
| 6     | Document 6    |
| 7     | Document 7    |
| 8     | Document 8    |
| 9     | Document 9    |
| 10    | Document 10   |
| 11    | Document 11   |
| 12    | Document 12   |
| 13    | Document 13   |
| 14    | Document 14   |
| 15    | Document 15   |
| 16    | Document 16   |
| 17    | Document 17   |

**Document 1 Content:**

**Category:** children

**Text:** the house Jim says he rum ; and as he spoke he reeled a little and caught himself with one hand against the wall Are you hurt? cried I Rum he repeated I must get away from here Rum! Rum! I ran to fetch it but I was quite unsteadied by all that had fallen out and I broke one glass and fouled the tap and while I was still getting in my own way I heard a loud fall in the parlour and running in beheld the captain lying full length upon the floor At the same instant my mother alarmed by the cries and fighting came running downstairs to help me Between us we raised his head He was breathing very loud and hard but his eyes were closed and his face a horrible colour Dear deary me cried my mother what a disgrace upon the house! And your poor father sick! In the mean-

The second example demonstrates how to quickly visualize your corpus with **Word Cloud**. We could connect **Word Cloud** directly to **Corpus**, but instead, we decided to apply some preprocessing with **Preprocess Text**. We are again working with *book-excerpts.tab*. We've put all text to lowercase, tokenized (split) the text to words only, filtered out English stopwords and selected 100 most frequent tokens.

The screenshot displays the Orange Data Mining software interface. At the top, a workflow is visible with three widgets: 'Corpus', 'Preprocess Text', and 'Word Cloud', connected by arrows. The 'Preprocess Text' widget is selected, showing its configuration panel on the left. The 'Word Cloud' widget is also selected, showing its configuration panel on the right.

**Preprocess Text Widget Configuration:**

- Info:** Document count: 140, Total tokens: 13541, Total types: 100.
- Transformation:**
  - ☒ Lowercase
  - ☐ Remove accents
  - ☐ Par
- Tokenization:**
  - ☐ Word & Punctuation
  - ☐ Whitespace
  - ☐ Sentence
  - ☒ Regexp (Pattern: \w+)
  - ☐ Tweet
- Normalization:** [disabled]
- Filtering:**
  - ☒ Stopwords (English, (none))
  - ☐ Lexicon ((none))
  - ☐ Regexp ((.\*[0-9]{2,.}\*)(^\w\$))
  - ☐ Document frequency (0,02, 0,95)

**Word Cloud Widget Configuration:**

- Info:** 0 words in a topic, 140 documents with 100 words.
- Cloud preferences:**
  - ☒ Color words
  - Words tilt: no
  - Regenerate word cloud
- Words & weights:**

| Weight | Word   |
|--------|--------|
| 846    | said   |
| 437    | one    |
| 340    | little |
| 289    | like   |
| 273    | could  |
| 264    | would  |
| 252    | time   |
| 228    | know   |
| 207    | man    |

The Word Cloud widget displays a visualization of the words and their weights, showing a dense cluster of words with 'said' being the most prominent.

