Corpus Viewer

Displays corpus content.

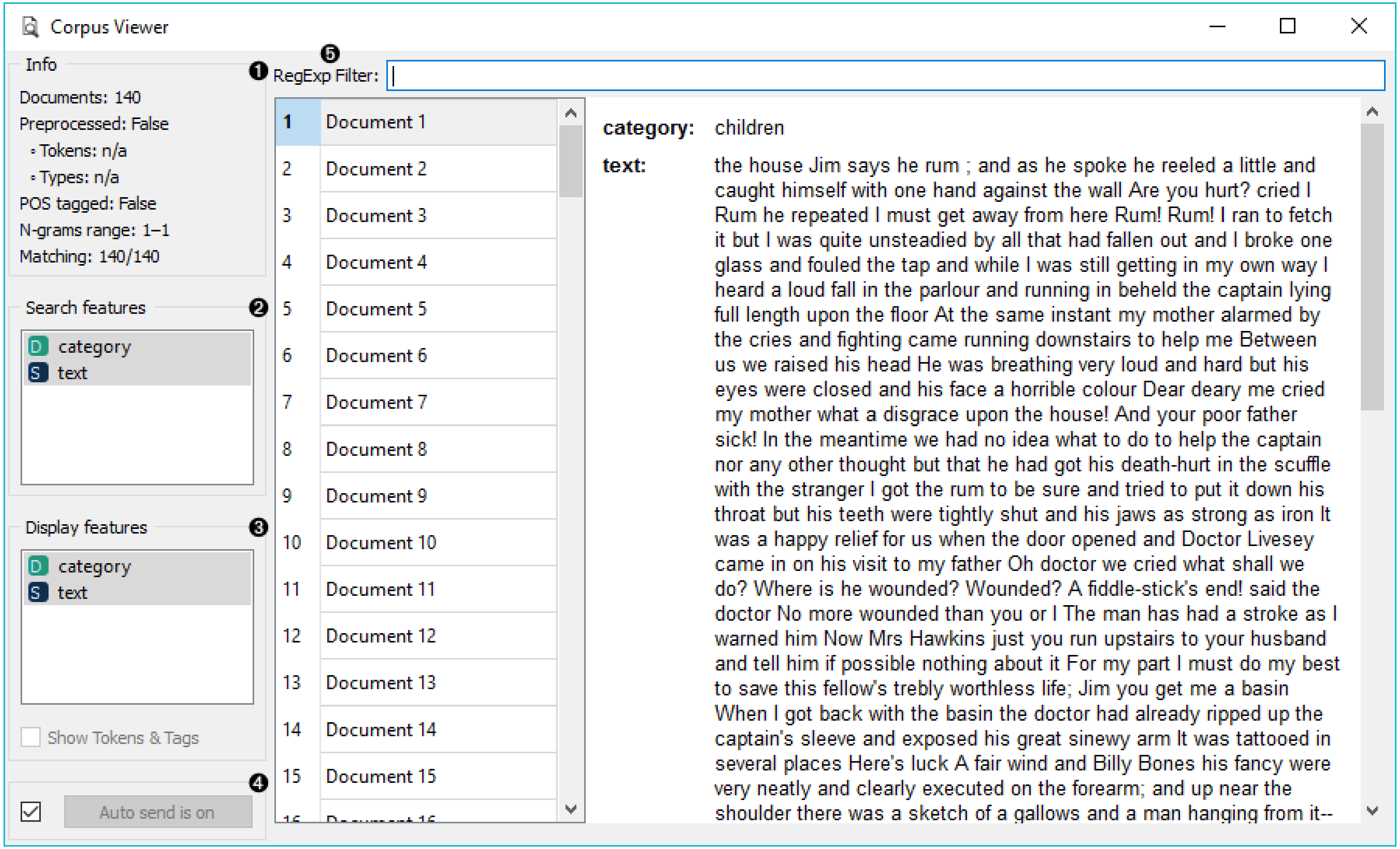
Inputs

- Corpus: A collection of documents.

Outputs

- Corpus: Documents containing the queried word.

Corpus Viewer is meant for viewing text files (instances of Corpus). It will always output an instance of corpus. If *RegExp* filtering is used, the widget will output only matching documents.



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

D category

S text

Display features

D category

S text

☐ Show Tokens & Tags

☑ Auto send is on

RegExp Filter:

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

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 meantime we had no idea what to do to help the captain nor any other thought but that he had got his death-hurt in the scuffle with the stranger I got the rum to be sure and tried to put it down his throat but his teeth were tightly shut and his jaws as strong as iron It was a happy relief for us when the door opened and Doctor Livesey came in on his visit to my father Oh doctor we cried what shall we do? Where is he wounded? Wounded? A fiddle-stick's end! said the doctor No more wounded than you or I The man has had a stroke as I warned him Now Mrs Hawkins just you run upstairs to your husband and tell him if possible nothing about it For my part I must do my best to save this fellow's trebly worthless life; Jim you get me a basin When I got back with the basin the doctor had already ripped up the captain's sleeve and exposed his great sinewy arm It was tattooed in several places Here's luck A fair wind and Billy Bones his fancy were very neatly and clearly executed on the forearm; and up near the shoulder there was a sketch of a gallows and a man hanging from it--

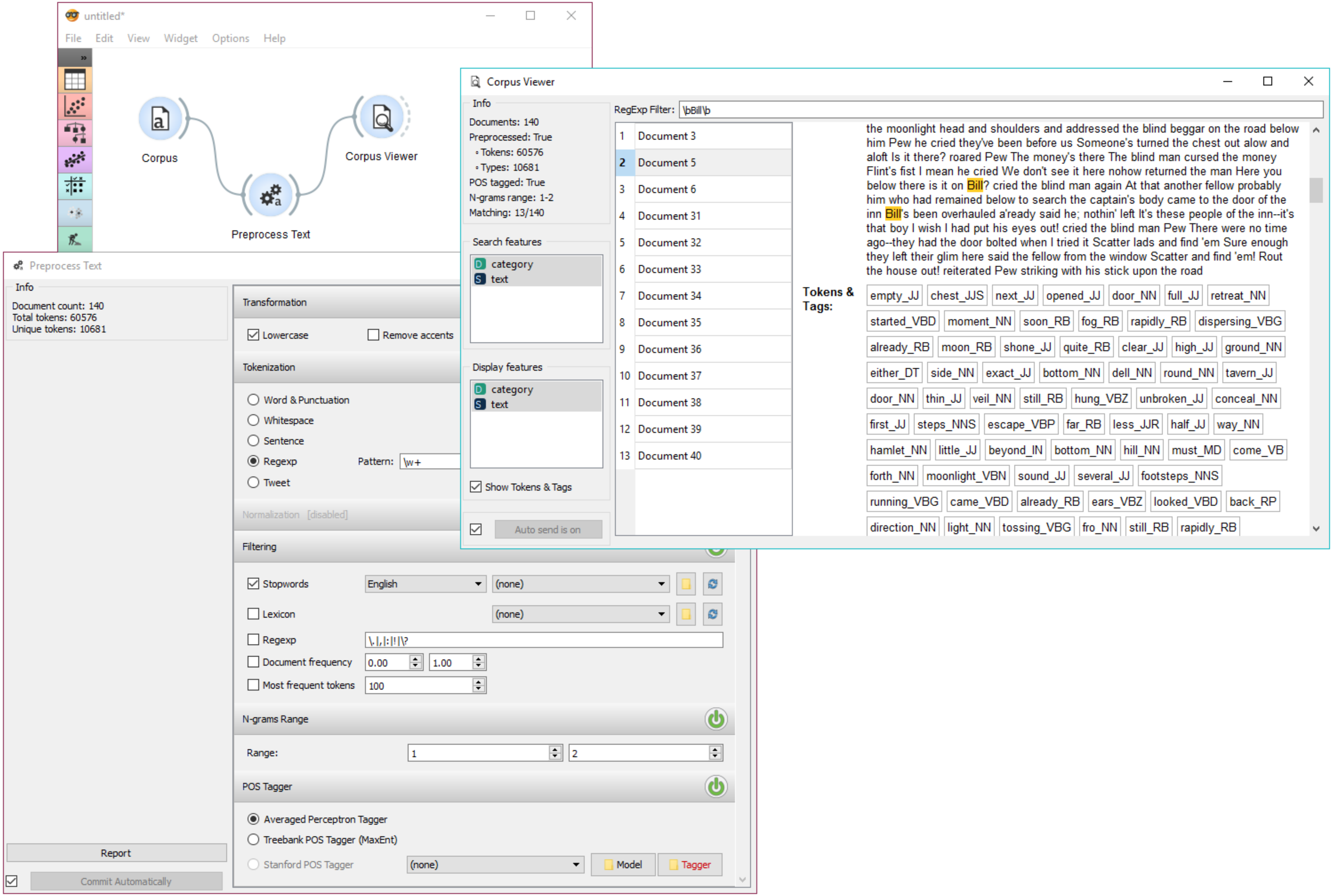
1. *Information*:
   - *Documents*: number of documents on the input
   - *Preprocessed*: if preprocessor is used, the result is True, else False. Reports also on the number of tokens and types (unique tokens).
   - *POS tagged*: if POS tags are on the input, the result is True, else False.
   - *N-grams range*: if N-grams are set in Preprocess Text, results are reported, default is 1-1 (one-grams).
   - *Matching*: number of documents matching the *RegExp Filter*. All documents are output by default.
2. *RegExp Filter*: Python regular expression for filtering documents. By default no documents are filtered (entire corpus is on the output).
3. *Search Features*: features by which the RegExp Filter is filtering. Use Ctrl (Cmd) to select multiple features.

4. *Display Features*: features that are displayed in the viewer. Use Ctrl (Cmd) to select multiple features.
5. *Show Tokens & Tags*: if tokens and POS tag are present on the input, you can check this box to display them.
6. If *Auto commit is on*, changes are communicated automatically. Alternatively press *Commit*.

Example

Corpus Viewer can be used for displaying all or some documents in corpus. In this example, we will first load *book-excerpts.tab*, that already comes with the add-on, into Corpus widget. Then we will preprocess the text into words, filter out the stopwords, create bi-grams and add POS tags (more on preprocessing in Preprocess Text. Now we want to see the results of preprocessing. In *Corpus Viewer* we can see, how many unique tokens we got and what they are (tick *Show Tokens & Tags*). Since we used also POS tagger to show part-of-speech labels, they will be displayed alongside tokens underneath the text.

Now we will filter out just the documents talking about a character Bill. We use regular expression *\bBill\b* to find the documents containing only the word Bill. You can output matching or non-matching documents, view them in another *Corpus Viewer* or further analyse them.



Corpus Viewer

Info

Documents: 140
Preprocessed: True
◦ Tokens: 60576
◦ Types: 10681
POS tagged: True
N-grams range: 1-2
Matching: 13/140

Search features

- D category
- S text

Display features

- D category
- S text

☑ Show Tokens & Tags

☑ Auto send is on

RegExp Filter: \bBill\b

1	Document 3
2	Document 5
3	Document 6
4	Document 31
5	Document 32
6	Document 33
7	Document 34
8	Document 35
9	Document 36
10	Document 37
11	Document 38
12	Document 39
13	Document 40

the moonlight head and shoulders and addressed the blind beggar on the road below him Pew he cried they've been before us Someone's turned the chest out alow and aloft Is it there? roared Pew The money's there The blind man cursed the money Flint's fist I mean he cried We don't see it here nohow returned the man Here you below there is it on Bill? cried the blind man again At that another fellow probably him who had remained below to search the captain's body came to the door of the inn Bill's been overhauled a'ready said he; nothin' left It's these people of the inn--it's that boy I wish I had put his eyes out! cried the blind man Pew There were no time ago--they had the door bolted when I tried it Scatter lads and find 'em Sure enough they left their glim here said the fellow from the window Scatter and find 'em! Rout the house out! reiterated Pew striking with his stick upon the road

Tokens & Tags:

empty_JJ chest_JJS next_JJ opened_JJ door_NN full_JJ retreat_NN
started_VBD moment_NN soon_RB fog_RB rapidly_RB dispersing_VBG
already_RB moon_RB shone_JJ quite_RB clear_JJ high_JJ ground_NN
either_DT side_NN exact_JJ bottom_NN dell_NN round_NN tavern_JJ
door_NN thin_JJ veil_NN still_RB hung_VBZ unbroken_JJ conceal_NN
first_JJ steps_NNS escape_VBP far_RB less_JJR half_JJ way_NN
hamlet_NN little_JJ beyond_IN bottom_NN hill_NN must_MD come_VB
forth_NN moonlight_VBN sound_JJ several_JJ footsteps_NNS
running_VBG came_VBD already_RB ears_VBZ looked_VBD back_RP
direction_NN light_NN tossing_VBG fro_NN still_RB rapidly_RB

Preprocess Text

Info

Document count: 140
Total tokens: 60576
Unique tokens: 10681

Transformation

☑ Lowercase ☐ Remove accents

Tokenization

○ Word & Punctuation
○ Whitespace
○ Sentence
● Regexp Pattern: \w+
○ Tweet

Normalization [disabled]

Filtering

☑ Stopwords English (none)
☐ Lexicon (none)
☐ Regexp \.|,|:|!|\?
☐ Document frequency 0.00 1.00
☐ Most frequent tokens 100

N-grams Range

Range: 1 2

POS Tagger

● Averaged Perceptron Tagger
○ Treebank POS Tagger (MaxEnt)
○ Stanford POS Tagger (none) Model Tagger

Report

☑ Commit Automatically