Pre-processing and topic modelling

- Regular expressions (regex) is used to locate needed pieces of text.

- If you recognize the pattern of how to identify your needed text - you can create rules for search.
  - Pattern examples: words separated by space or comma or dot… Each new sentence starts with capital letter… If XML tag is opened, it needs to be closed

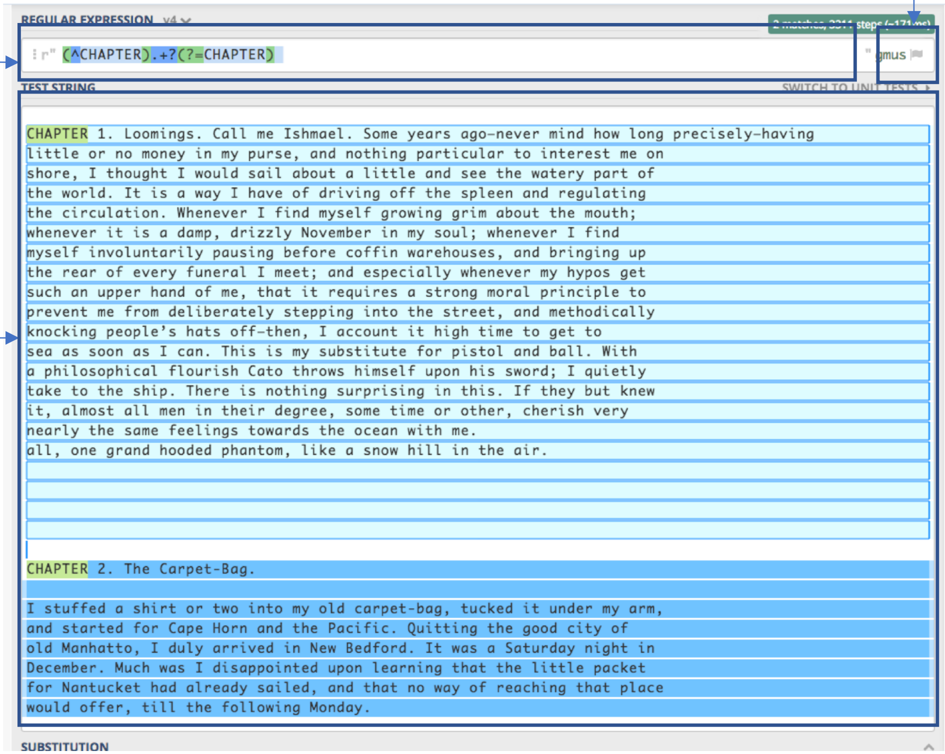
Try on

- https://regex101.com/

Set flags for expression

Add regular expression

Copy some text



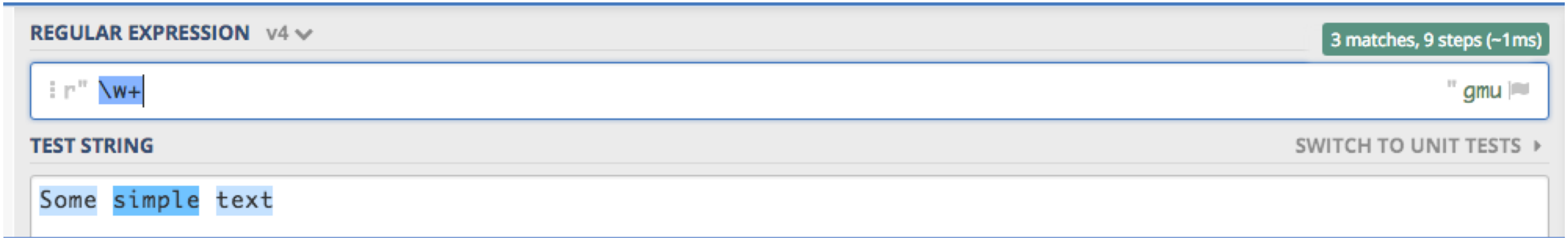
```
r" (^CHAPTER).+?(?=CHAPTER) " gmus
```

```
CHAPTER 1. Loomings. Call me Ishmael. Some years ago-never mind how long precisely-having
little or no money in my purse, and nothing particular to interest me on
shore, I thought I would sail about a little and see the watery part of
the world. It is a way I have of driving off the spleen and regulating
the circulation. Whenever I find myself growing grim about the mouth;
whenever it is a damp, drizzly November in my soul; whenever I find
myself involuntarily pausing before coffin warehouses, and bringing up
the rear of every funeral I meet; and especially whenever my hypos get
such an upper hand of me, that it requires a strong moral principle to
prevent me from deliberately stepping into the street, and methodically
knocking people's hats off-then, I account it high time to get to
sea as soon as I can. This is my substitute for pistol and ball. With
a philosophical flourish Cato throws himself upon his sword; I quietly
take to the ship. There is nothing surprising in this. If they but knew
it, almost all men in their degree, some time or other, cherish very
nearly the same feelings towards the ocean with me.
all, one grand hooded phantom, like a snow hill in the air.




CHAPTER 2. The Carpet-Bag.

I stuffed a shirt or two into my old carpet-bag, tucked it under my arm,
and started for Cape Horn and the Pacific. Quitting the good city of
old Manhatto, I duly arrived in New Bedford. It was a Saturday night in
December. Much was I disappointed upon learning that the little packet
for Nantucket had already sailed, and that no way of reaching that place
would offer, till the following Monday.
```

Metacharacters

\w	- any letter
\w+	- any word
\d	- any number
\s	- any whitespace
\S	- any character non-space
.	- every symbol except end of line
\n	- end of line

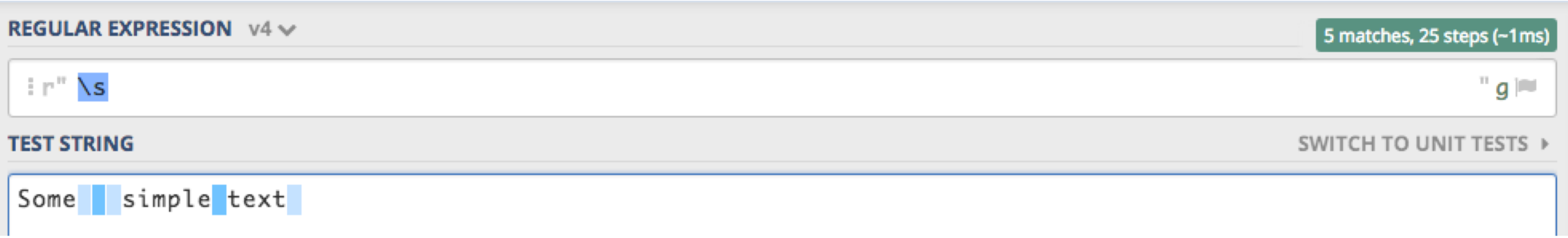
Matches words



```
r" \w+ " gmu
```

```
Some simple text
```

Matches spaces



```
r" \s " g
```

```
Some   simple text 
```

Metacharacters

[ ] - everything used within brackets directly relates to search of the characters

| - OR syntax

^ - starts matching from the beginning of the text element

\A - starts matching from the beginning of the text element, but not affected by newline character (\n)

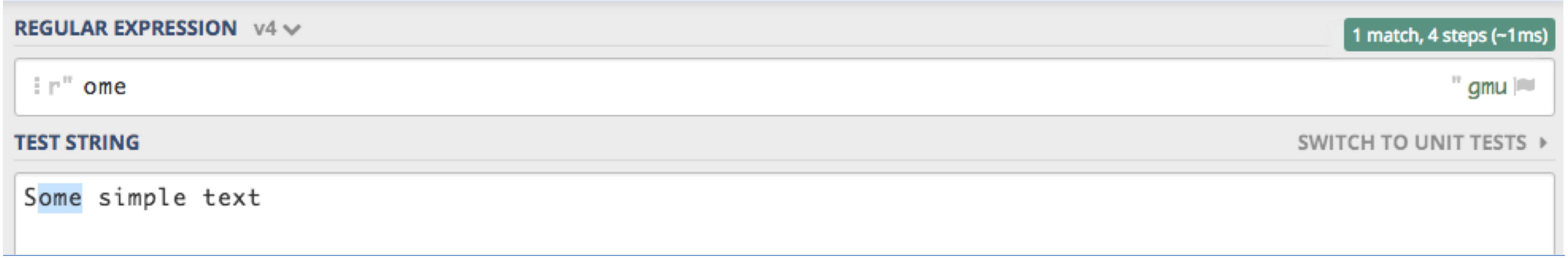
$ - starts matching from the end of the text element

\Z - Matches only at the end of the string, but not affected by the newline character

\b – for matching only separated, by space or other character like comma, dot, dash, words.

\B – matching only sequences that exists within words, but doesn't start or end exactly with

Matches and returns exact phrase



REGULAR EXPRESSION v4

1 match, 4 steps (~1ms)

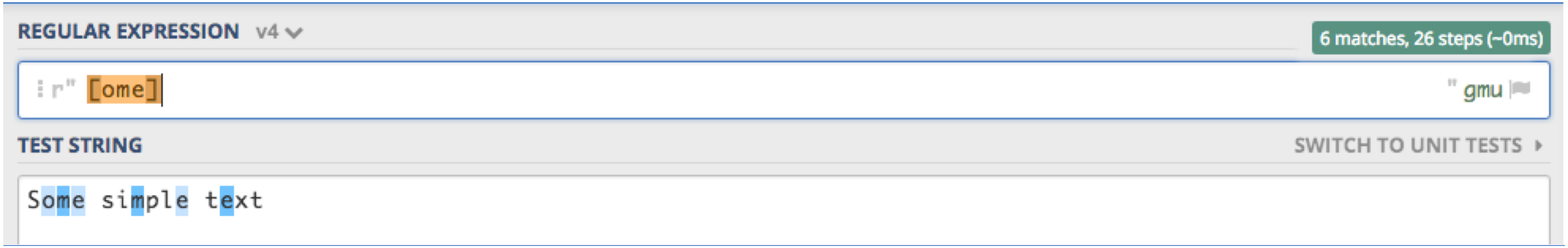
```
r" ome " gmu
```

TEST STRING

SWITCH TO UNIT TESTS

```
Some simple text
```

Matches and returns any of the symbol within brackets



REGULAR EXPRESSION v4

6 matches, 26 steps (~0ms)

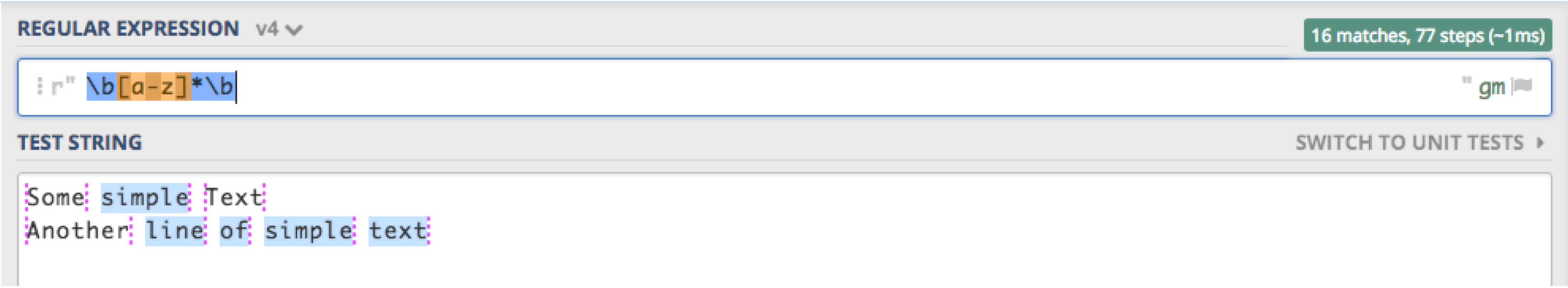
```
r" [ome] " gmu
```

TEST STRING

SWITCH TO UNIT TESTS

```
Some simple text
```

Matching within each word

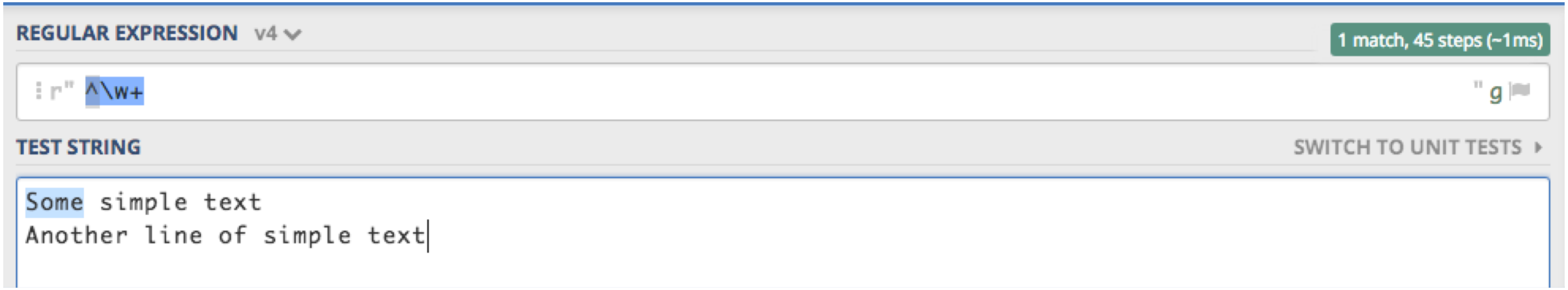


16 matches, 77 steps (~1ms)

```
r" \b[a-z]*\b " gm
```

```
Some simple Text
Another line of simple text
```

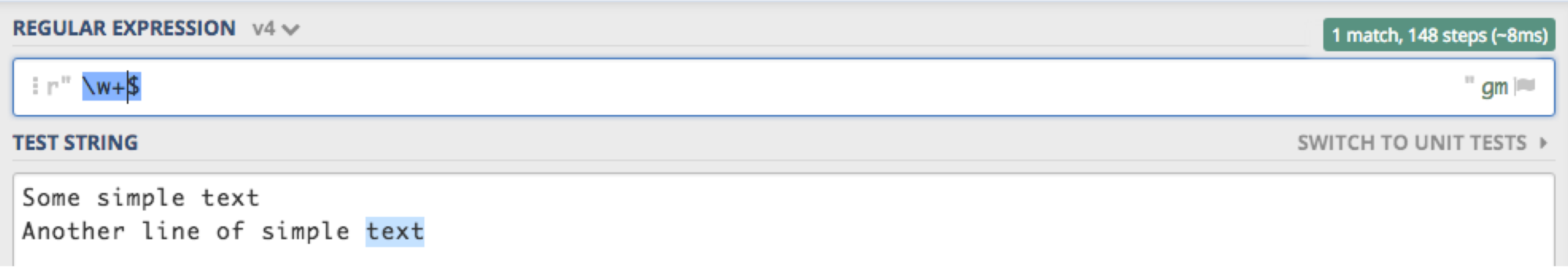
Starts matching from the beginning of the text



```
r" ^\w+ " g
```

```
Some simple text
Another line of simple text
```

Starts matching from the end of the text



```
r" \w+$ " gm
```

```
Some simple text
Another line of simple text
```

Flags

S – are use with .(dot) to match any character including newline symbol

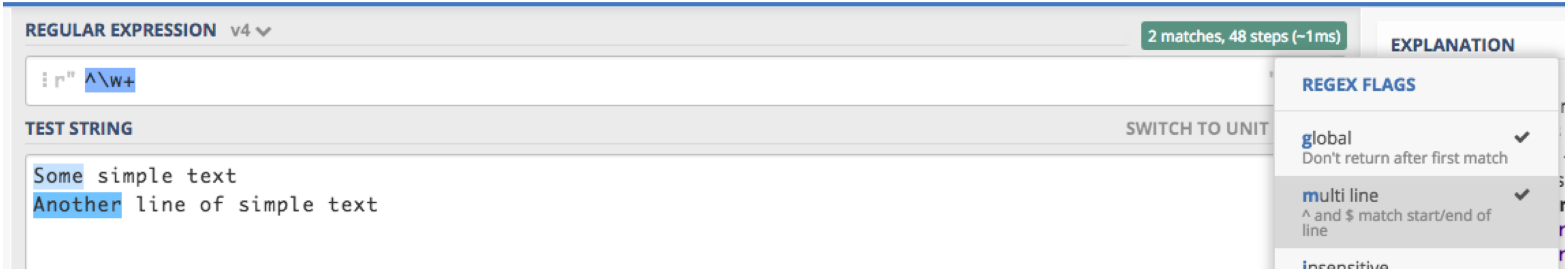
I – case sensitive matching

L – for matching non English alphabet text

M – used with ^ and $ to treat whole text as single text for each new line

X – allows to use white spaces and commenting for better REGEX readability

With multiline flag it matches from the beginning of each line



```
r" ^\w+
```

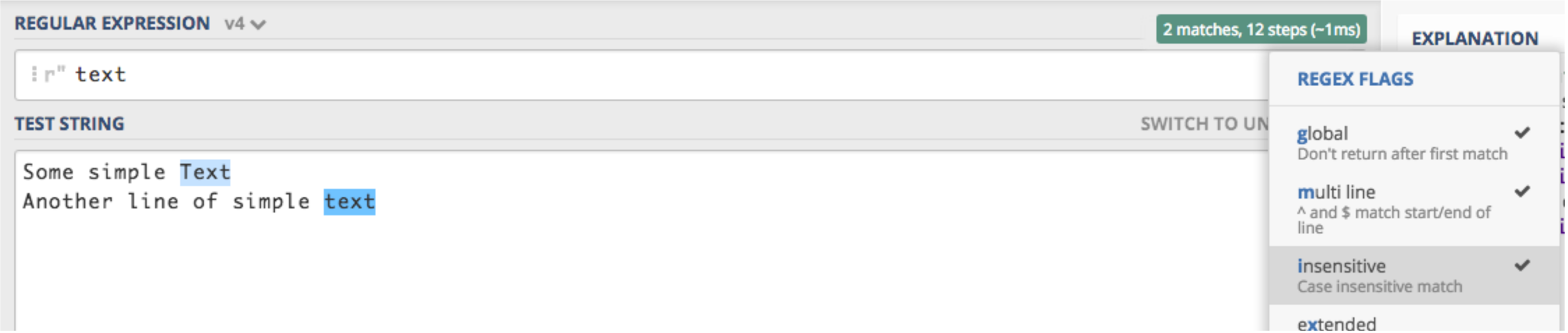
```
Some simple text
Another line of simple text
```

global ✓
Don't return after first match

multi line ✓
^ and $ match start/end of line

insensitive

With insensitive flag it matches and lowercase and uppercase letters



```
r" text
```

```
Some simple Text
Another line of simple text
```

global ✓
Don't return after first match

multi line ✓
^ and $ match start/end of line

insensitive ✓
Case insensitive match

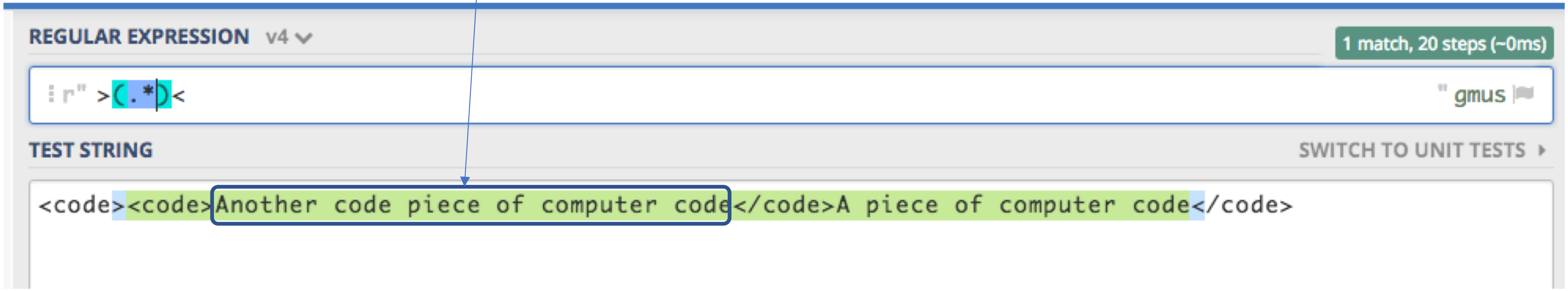
extended

Non-Greedy matching

{m,n}? – defining exact start and ending of elements to match

*? – stop matching at first occurrence of the pattern

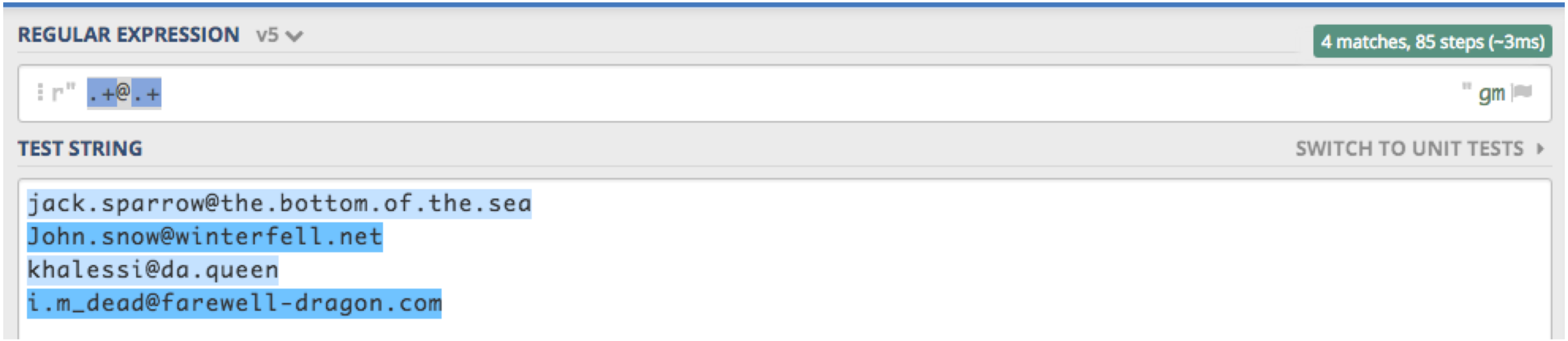
We want only this piece



```
r" >(.*)< " gmus
```

```
<code><code>Another code piece of computer code</code>A piece of computer code</code>
```

Exercise 1: email address match



4 matches, 85 steps (~3ms)

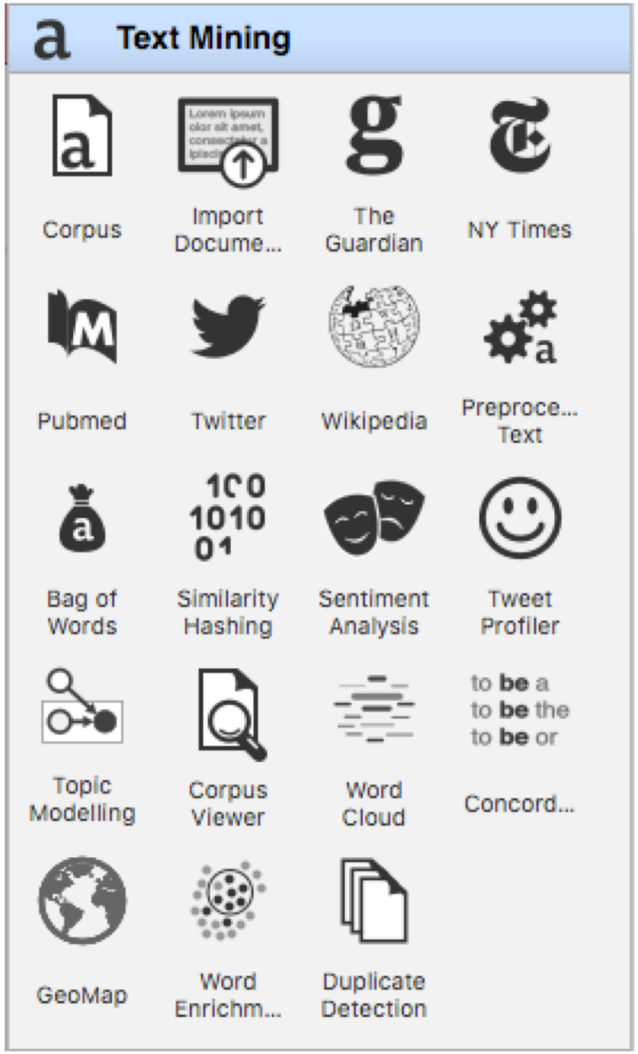
```
r" .+@.+ " gm
```

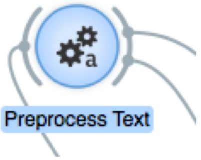
```
jack.sparrow@the.bottom.of.the.sea
John.snow@winterfell.net
khalessi@da.queen
i.m_dead@farewell-dragon.com
```

https://regex101.com/r/qrI054/7

Topic modeling

1st option: Text Mining add-on





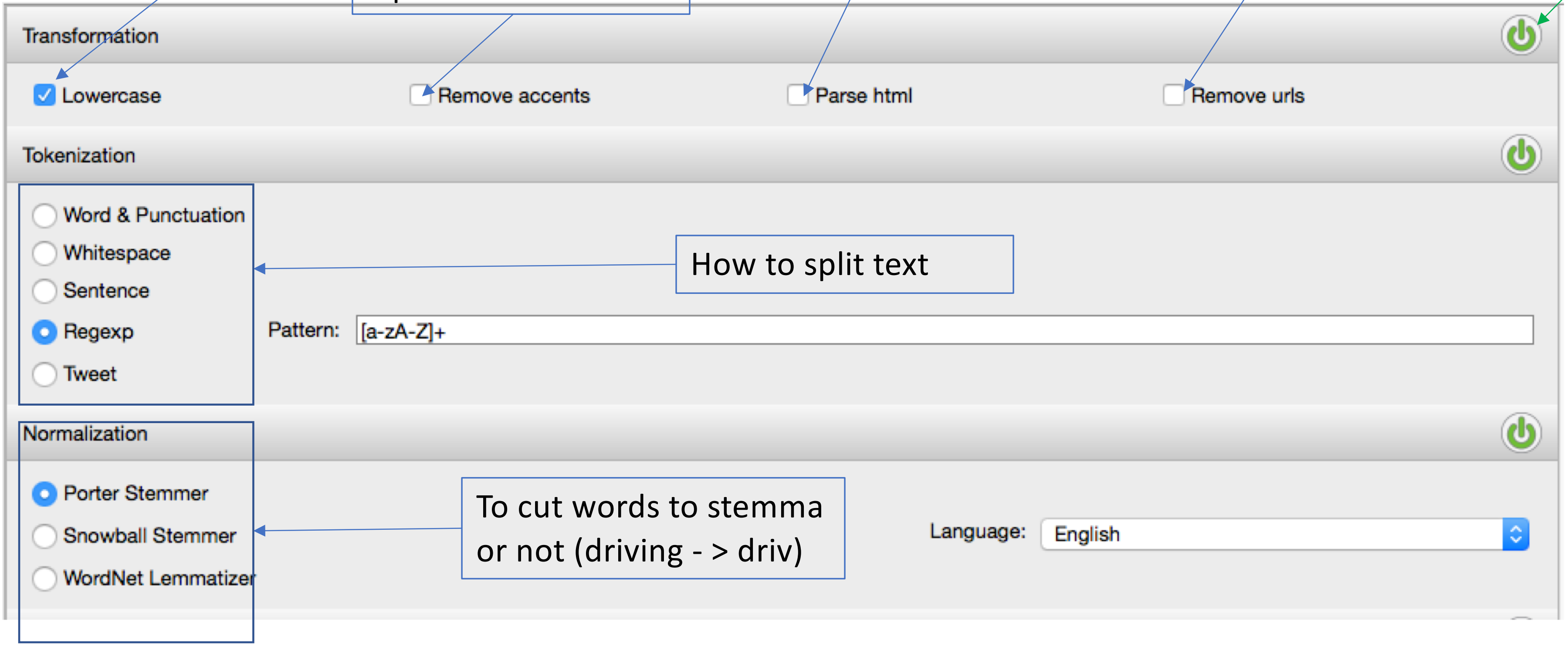
Switch if function needs to be performed

Transform letter to lowercase

Remove non-English alphabet letters

Remove tags, leave only visible text

Remove internet addresses



Transformation

☑ Lowercase □ Remove accents □ Parse html □ Remove urls

Tokenization

○ Word & Punctuation
○ Whitespace
○ Sentence
● Regexp
○ Tweet

Pattern: [a-zA-Z]+

How to split text

Normalization

● Porter Stemmer
○ Snowball Stemmer
○ WordNet Lemmatizer

Language: English

To cut words to stemma or not (driving - > driv)

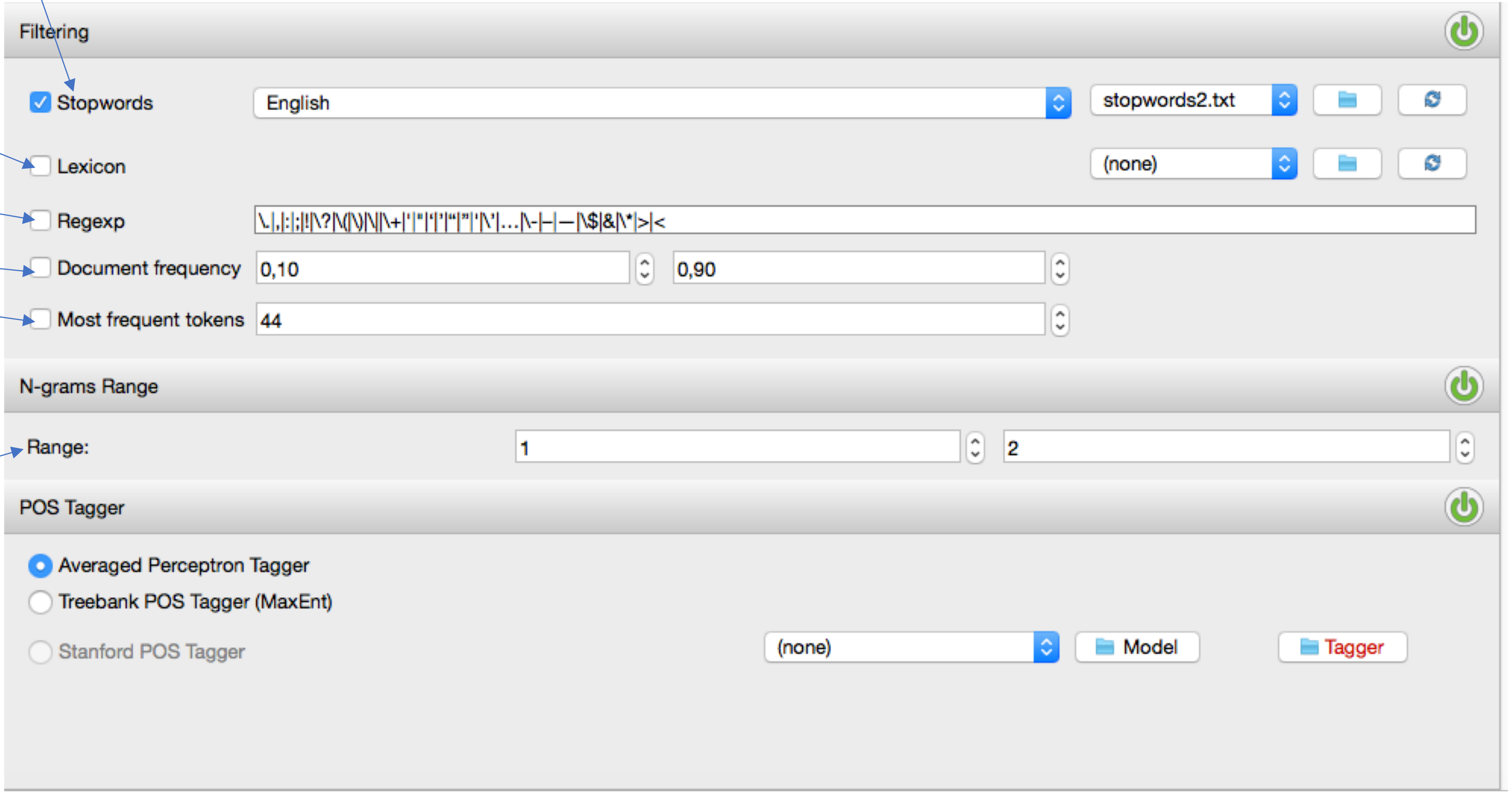
Removing words by defined frequent list of words (and, but not…)

Removing words by supplied dictionary

… by regular express.

… by how frequent words are within the document

To use only single word in analysis, or phrases of 1 or more



Filtering

☑ Stopwords: English; stopwords2.txt

□ Lexicon: (none)

□ Regexp: `\.|,|:|;|!|\?|\(|\)|\||\+|'|"|‘|’|“|”|'|\'|…|\-|–|—|\$|&|\*|>|<`

□ Document frequency: 0,10 – 0,90

□ Most frequent tokens: 44

N-grams Range

Range: 1 – 2

POS Tagger

● Averaged Perceptron Tagger

○ Treebank POS Tagger (MaxEnt)

○ Stanford POS Tagger: (none)

Topic Modeling

Twitter API key

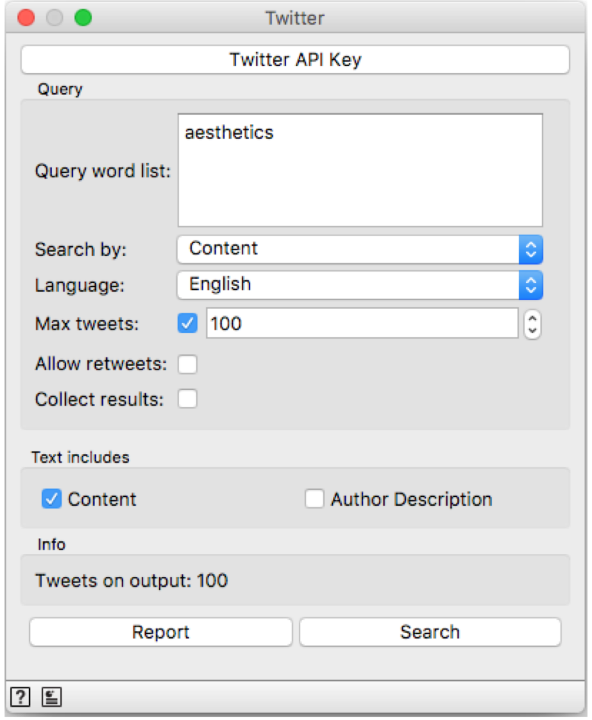
KEY:

nU02XOBxxuWuvHJiJ42MR6bsW

SECRET:

NFkMtuu9XbekfkbvbW5olzK2QEiVpXCDLM5YWPFYmLYdbHpBXy

Latent Dirichlet Allocation (LDA)



Twitter

Twitter API Key

Query

Query word list: aesthetics

Search by: Content

Language: English

Max tweets: ☑ 100

Allow retweets: □

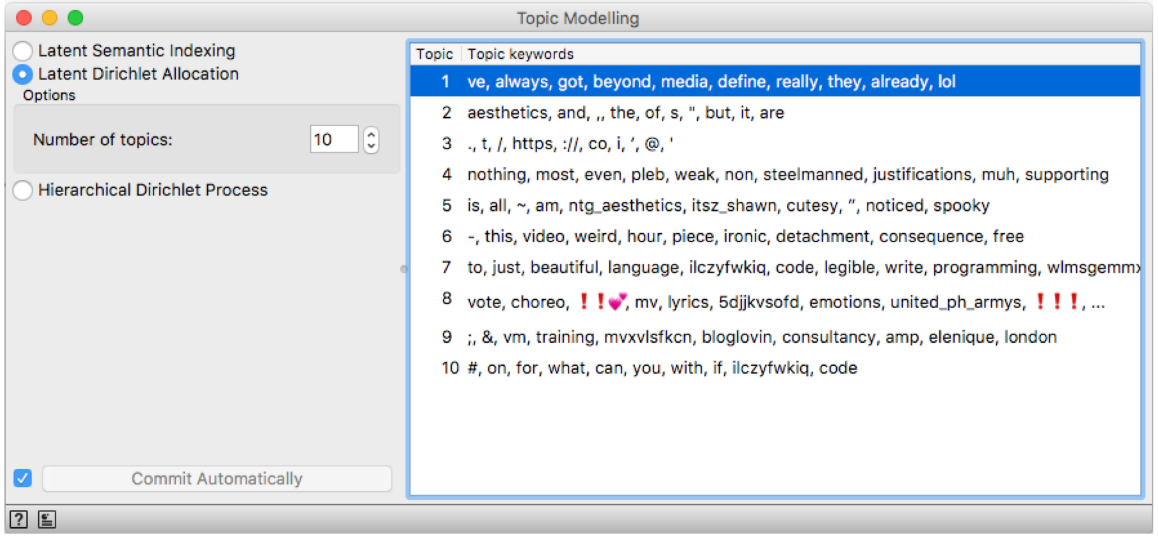
Collect results: □

Text includes

☑ Content □ Author Description

Info

Tweets on output: 100



Topic Modelling

○ Latent Semantic Indexing

● Latent Dirichlet Allocation

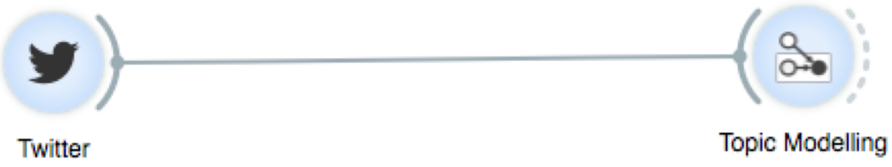
Options

Number of topics: 10

○ Hierarchical Dirichlet Process

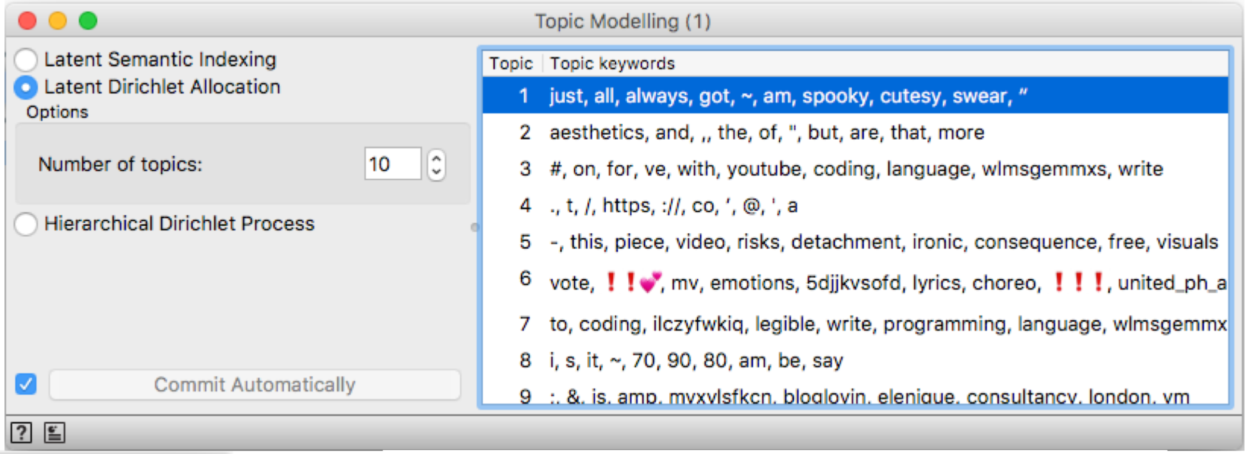
☑ Commit Automatically

Topic	Topic keywords
1	ve, always, got, beyond, media, define, really, they, already, lol
2	aesthetics, and, ,, the, of, s, ", but, it, are
3	., t, /, https, ://, co, i, ’, @, '
4	nothing, most, even, pleb, weak, non, steelmanned, justifications, muh, supporting
5	is, all, ~, am, ntg_aesthetics, itsz_shawn, cutesy, “, noticed, spooky
6	-, this, video, weird, hour, piece, ironic, detachment, consequence, free
7	to, just, beautiful, language, ilczyfwkiq, code, legible, write, programming, wlmsgemmx
8	vote, choreo, ❗❗💕, mv, lyrics, 5djjkvsofd, emotions, united_ph_armys, ❗❗❗, ...
9	;, &, vm, training, mvxvlsfkcn, bloglovin, consultancy, amp, elenique, london
10	#, on, for, what, can, you, with, if, ilczyfwkiq, code



Randomness in the results

Make two identical data flows with **Topic Modelling** and explore the results.

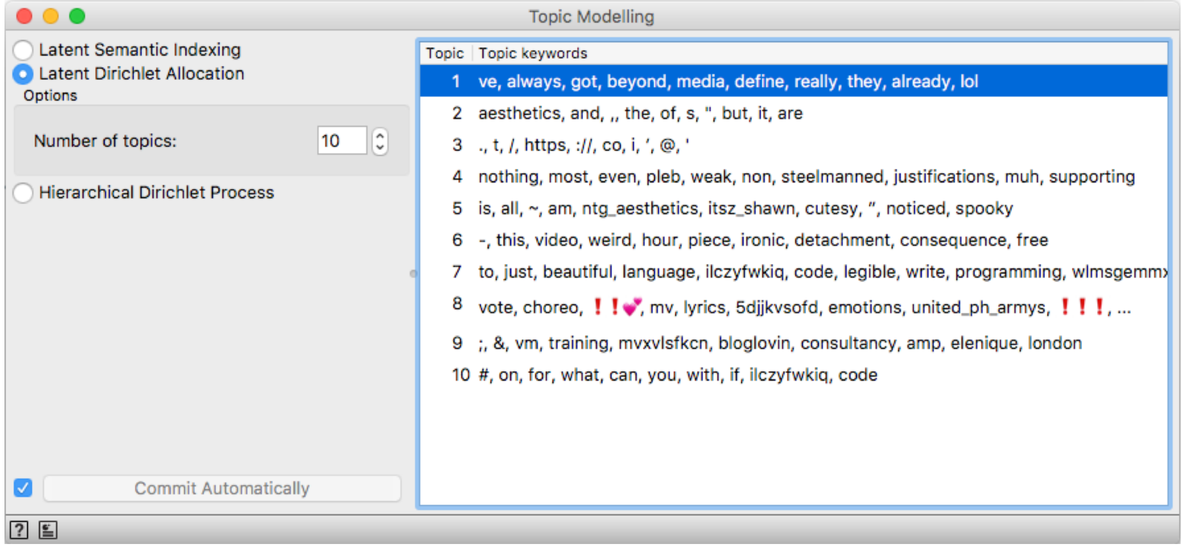


Topic Modelling (1)

○ Latent Semantic Indexing
● Latent Dirichlet Allocation
Options
Number of topics: 10
○ Hierarchical Dirichlet Process

☑ Commit Automatically

Topic	Topic keywords
1	just, all, always, got, ~, am, spooky, cutesy, swear, “
2	aesthetics, and, ,, the, of, ", but, are, that, more
3	#, on, for, ve, with, youtube, coding, language, wlmsgemmxs, write
4	., t, /, https, ://, co, ’, @, ', a
5	-, this, piece, video, risks, detachment, ironic, consequence, free, visuals
6	vote, ❗❗💕, mv, emotions, 5djjkvsofd, lyrics, choreo, ❗❗❗, united_ph_a
7	to, coding, ilczyfwkiq, legible, write, programming, language, wlmsgemmx
8	i, s, it, ~, 70, 90, 80, am, be, say
9	:, &, is, amp, mvxvlsfkcn, bloglovin, elenique, consultancy, london, vm

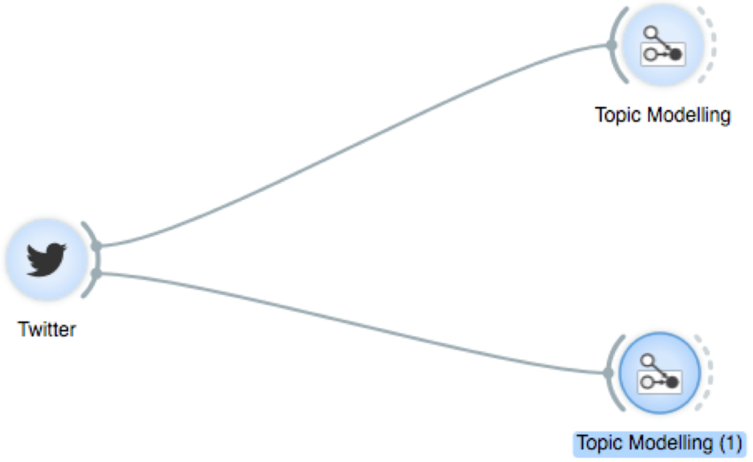


Topic Modelling

○ Latent Semantic Indexing
● Latent Dirichlet Allocation
Options
Number of topics: 10
○ Hierarchical Dirichlet Process

☑ Commit Automatically

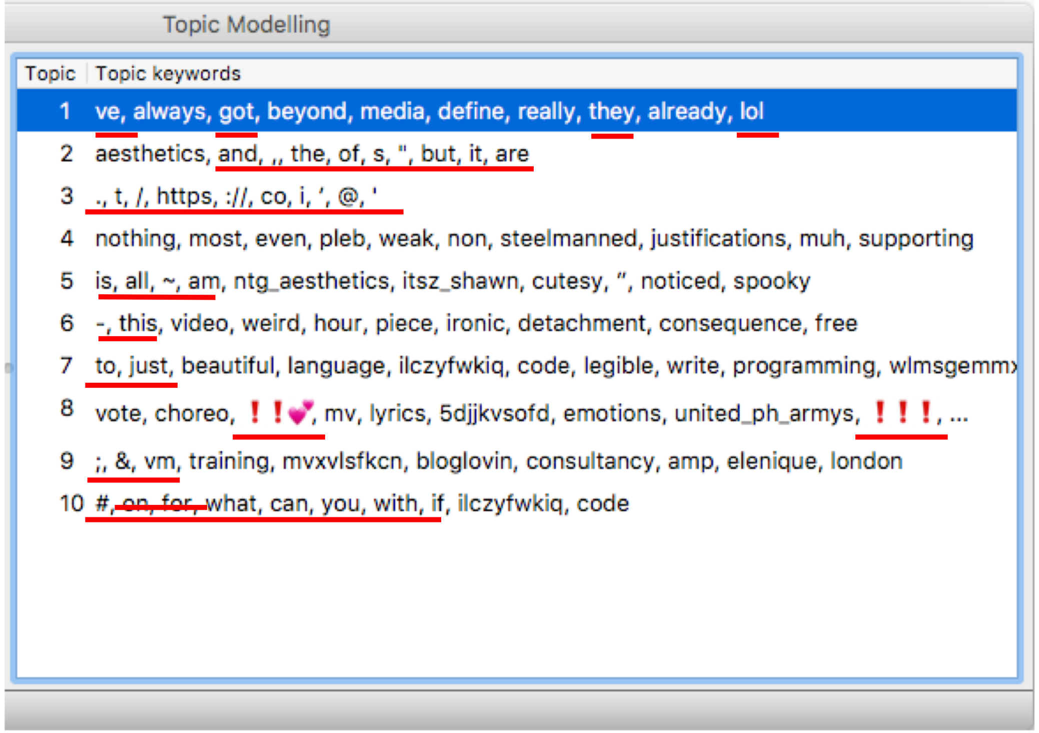
Topic	Topic keywords
1	ve, always, got, beyond, media, define, really, they, already, lol
2	aesthetics, and, ,, the, of, s, ", but, it, are
3	., t, /, https, ://, co, i, ’, @, '
4	nothing, most, even, pleb, weak, non, steelmanned, justifications, muh, supporting
5	is, all, ~, am, ntg_aesthetics, itsz_shawn, cutesy, “, noticed, spooky
6	-, this, video, weird, hour, piece, ironic, detachment, consequence, free
7	to, just, beautiful, language, ilczyfwkiq, code, legible, write, programming, wlmsgemmx
8	vote, choreo, ❗❗💕, mv, lyrics, 5djjkvsofd, emotions, united_ph_armys, ❗❗❗, ...
9	;, &, vm, training, mvxvlsfkcn, bloglovin, consultancy, amp, elenique, london
10	#, on, for, what, can, you, with, if, ilczyfwkiq, code



Lots of noise (garbage) in the data

- Preprocess it – lowercase, remove too usual, or unneeded words/text/numbers/symbols
- Check the results and adjust preprocessing

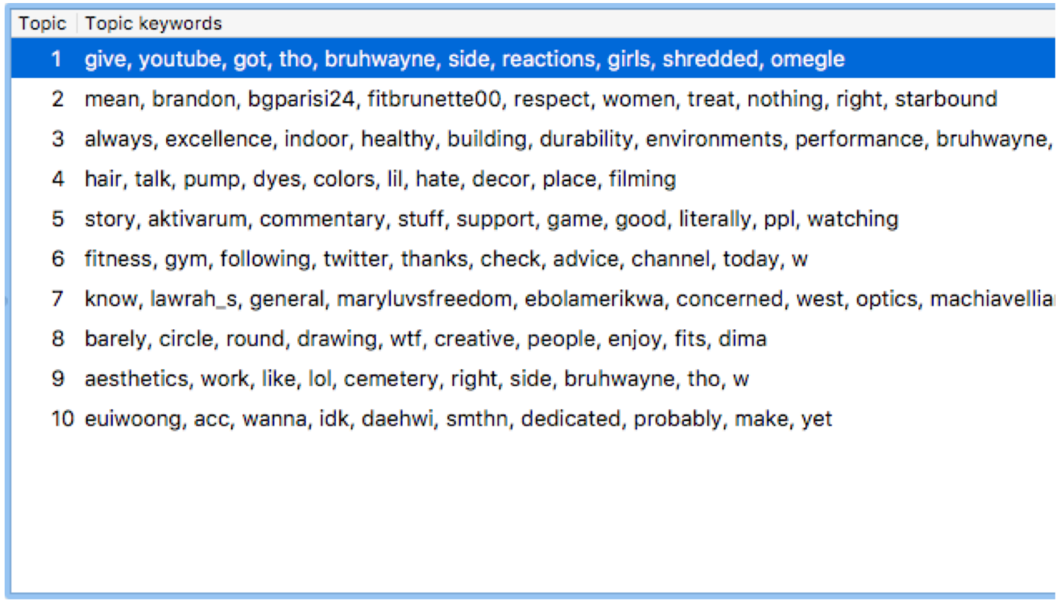
Before pre-processing



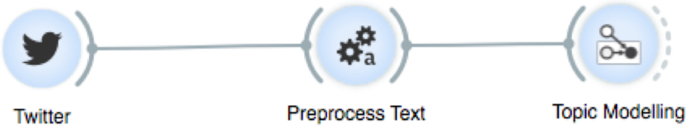
Topic Modelling

Topic	Topic keywords
1	ve, always, got, beyond, media, define, really, they, already, lol
2	aesthetics, and, ,, the, of, s, ", but, it, are
3	., t, /, https, ://, co, i, ’, @, '
4	nothing, most, even, pleb, weak, non, steelmanned, justifications, muh, supporting
5	is, all, ~, am, ntg_aesthetics, itsz_shawn, cutesy, “, noticed, spooky
6	-, this, video, weird, hour, piece, ironic, detachment, consequence, free
7	to, just, beautiful, language, ilczyfwkiq, code, legible, write, programming, wlmsgemm
8	vote, choreo, ❗❗💕, mv, lyrics, 5djjkvsofd, emotions, united_ph_armys, ❗❗❗, ...
9	;, &, vm, training, mvxvlsfkcn, bloglovin, consultancy, amp, elenique, london
10	#, ~~on, for,~~ what, can, you, with, if, ilczyfwkiq, code

After pre-processing



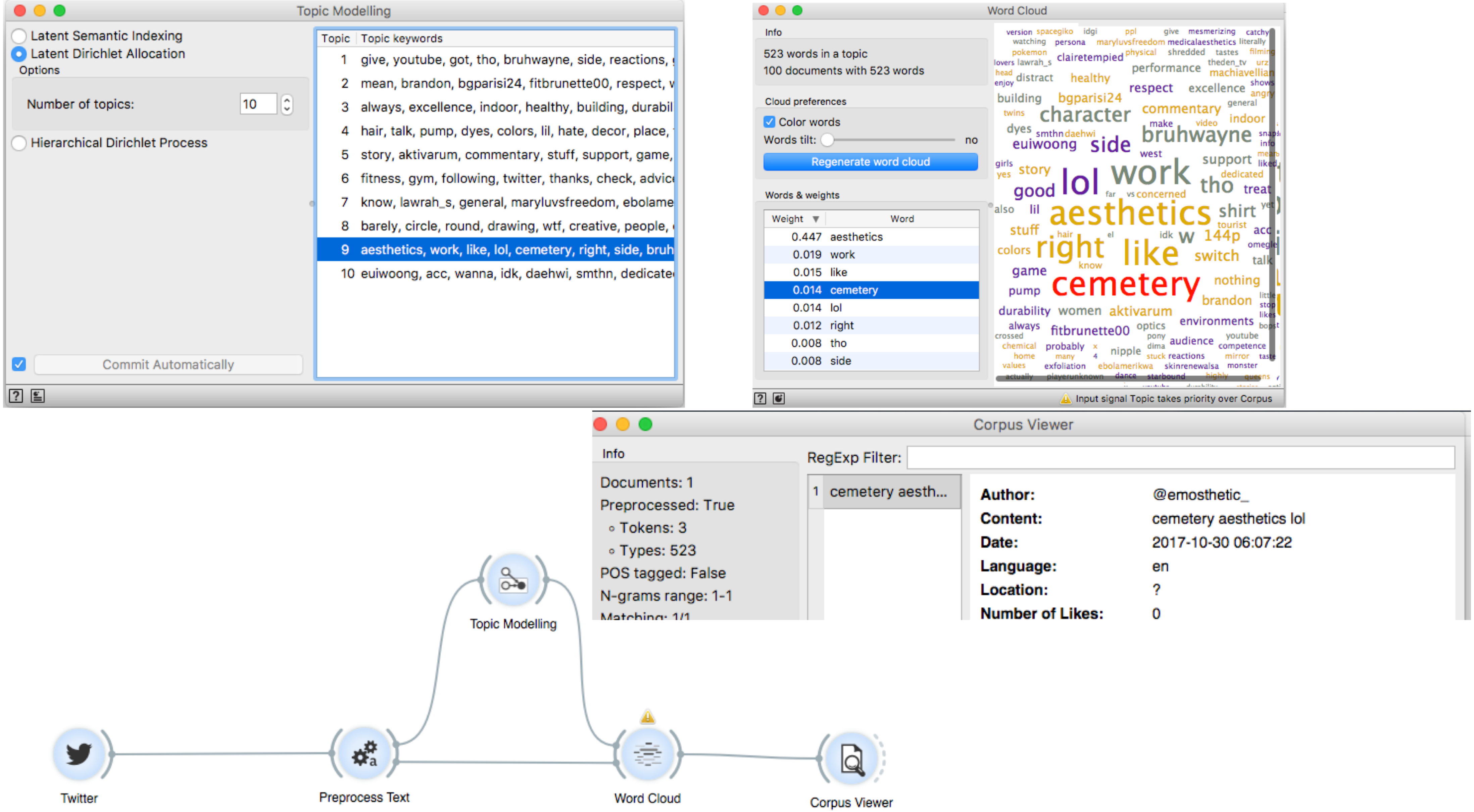
Topic	Topic keywords
1	give, youtube, got, tho, bruhwayne, side, reactions, girls, shredded, omegle
2	mean, brandon, bgparisi24, fitbrunette00, respect, women, treat, nothing, right, starbound
3	always, excellence, indoor, healthy, building, durability, environments, performance, bruhwayne,
4	hair, talk, pump, dyes, colors, lil, hate, decor, place, filming
5	story, aktivarum, commentary, stuff, support, game, good, literally, ppl, watching
6	fitness, gym, following, twitter, thanks, check, advice, channel, today, w
7	know, lawrah_s, general, maryluvsfreedom, ebolamerikwa, concerned, west, optics, machiavellia
8	barely, circle, round, drawing, wtf, creative, people, enjoy, fits, dima
9	aesthetics, work, like, lol, cemetery, right, side, bruhwayne, tho, w
10	euiwoong, acc, wanna, idk, daehwi, smthn, dedicated, probably, make, yet



Visualizing Topic Modelling results

Important issues

- Words that mostly contribute to the topic
- Documents that consists of these words
- Distribution of the topics (are there any topics that presented across all the documents)



Topic Modelling

○ Latent Semantic Indexing
● Latent Dirichlet Allocation
Options

Number of topics: 10

○ Hierarchical Dirichlet Process

☑ Commit Automatically

Topic	Topic keywords
1	give, youtube, got, tho, bruhwayne, side, reactions,
2	mean, brandon, bgparisi24, fitbrunette00, respect, w
3	always, excellence, indoor, healthy, building, durabil
4	hair, talk, pump, dyes, colors, lil, hate, decor, place,
5	story, aktivarum, commentary, stuff, support, game,
6	fitness, gym, following, twitter, thanks, check, advic
7	know, lawrah_s, general, maryluvsfreedom, ebolame
8	barely, circle, round, drawing, wtf, creative, people,
9	aesthetics, work, like, lol, cemetery, right, side, bruh
10	euiwoong, acc, wanna, idk, daehwi, smthn, dedicate

Word Cloud

Info

523 words in a topic
100 documents with 523 words

Cloud preferences

☑ Color words
Words tilt: no

Regenerate word cloud

Words & weights

Weight	Word
0.447	aesthetics
0.019	work
0.015	like
0.014	cemetery
0.014	lol
0.012	right
0.008	tho
0.008	side

⚠ Input signal Topic takes priority over Corpus

Corpus Viewer

Info

Documents: 1
Preprocessed: True
∘ Tokens: 3
∘ Types: 523
POS tagged: False
N-grams range: 1-1
Matching: 1/1

RegExp Filter:

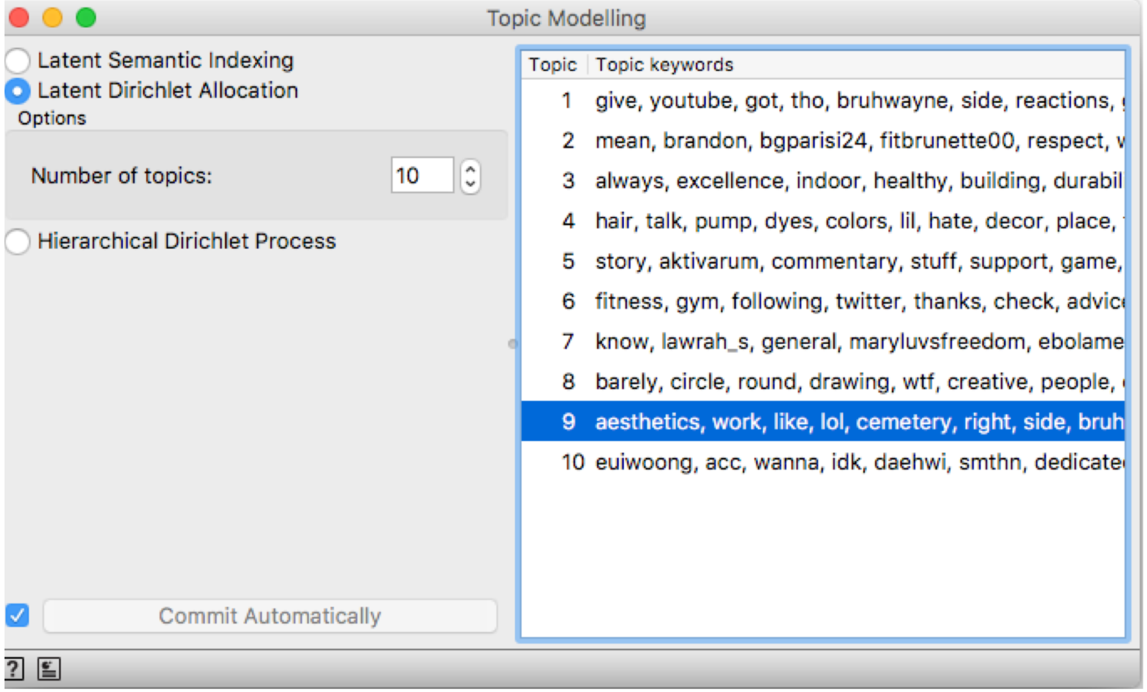
1 cemetery aesth...

Author:	@emosthetic_
Content:	cemetery aesthetics lol
Date:	2017-10-30 06:07:22
Language:	en
Location:	?
Number of Likes:	0

Results depends on the corpus and documents sizes
(bigger more reasonable results)

- Increase the amount of tweets and check the result

100 tweets



Topic Modelling

○ Latent Semantic Indexing

● Latent Dirichlet Allocation

Options

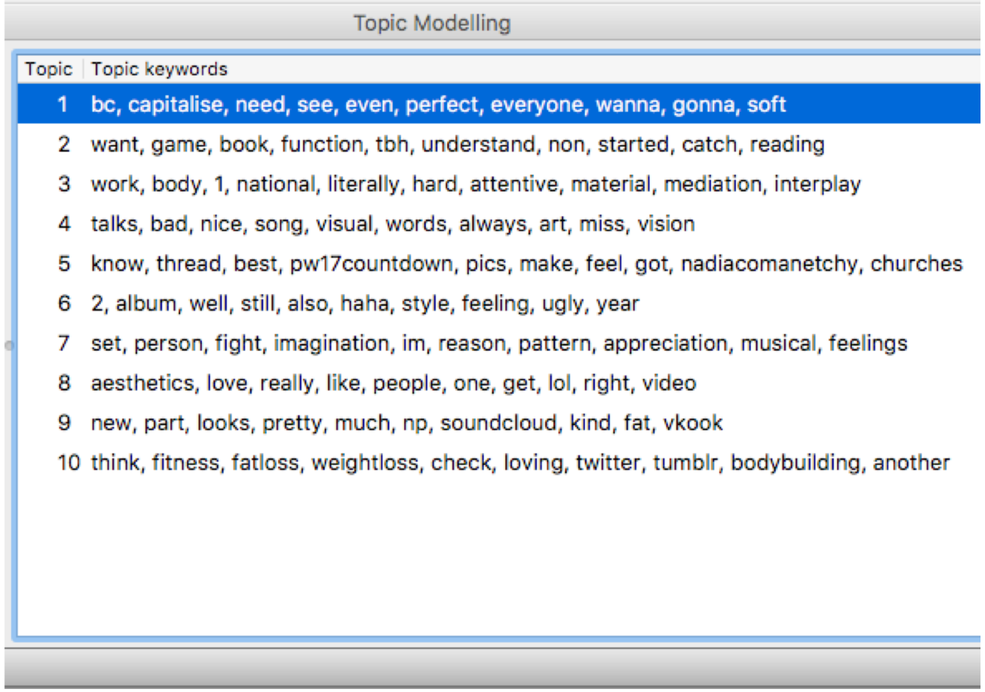
Number of topics: 10

○ Hierarchical Dirichlet Process

☑ Commit Automatically

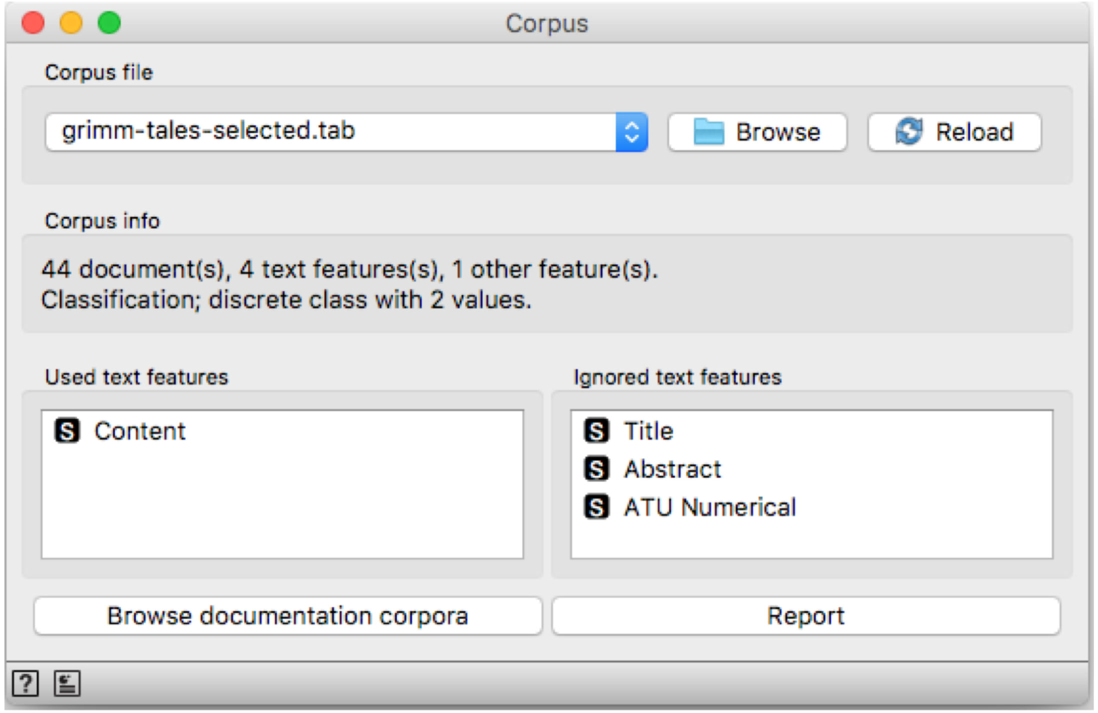
Topic	Topic keywords
1	give, youtube, got, tho, bruhwayne, side, reactions,
2	mean, brandon, bgparisi24, fitbrunette00, respect,
3	always, excellence, indoor, healthy, building, durabil
4	hair, talk, pump, dyes, colors, lil, hate, decor, place,
5	story, aktivarum, commentary, stuff, support, game,
6	fitness, gym, following, twitter, thanks, check, advic
7	know, lawrah_s, general, maryluvsfreedom, ebolame
8	barely, circle, round, drawing, wtf, creative, people,
9	aesthetics, work, like, lol, cemetery, right, side, bruh
10	euiwoong, acc, wanna, idk, daehwi, smthn, dedicate

1000 tweets



Topic Modelling

Topic	Topic keywords
1	bc, capitalise, need, see, even, perfect, everyone, wanna, gonna, soft
2	want, game, book, function, tbh, understand, non, started, catch, reading
3	work, body, 1, national, literally, hard, attentive, material, mediation, interplay
4	talks, bad, nice, song, visual, words, always, art, miss, vision
5	know, thread, best, pw17countdown, pics, make, feel, got, nadiacomanetchy, churches
6	2, album, well, still, also, haha, style, feeling, ugly, year
7	set, person, fight, imagination, im, reason, pattern, appreciation, musical, feelings
8	aesthetics, love, really, like, people, one, get, lol, right, video
9	new, part, looks, pretty, much, np, soundcloud, kind, fat, vkook
10	think, fitness, fatloss, weightloss, check, loving, twitter, tumblr, bodybuilding, another



Corpus

Corpus file

grimm-tales-selected.tab

Corpus info

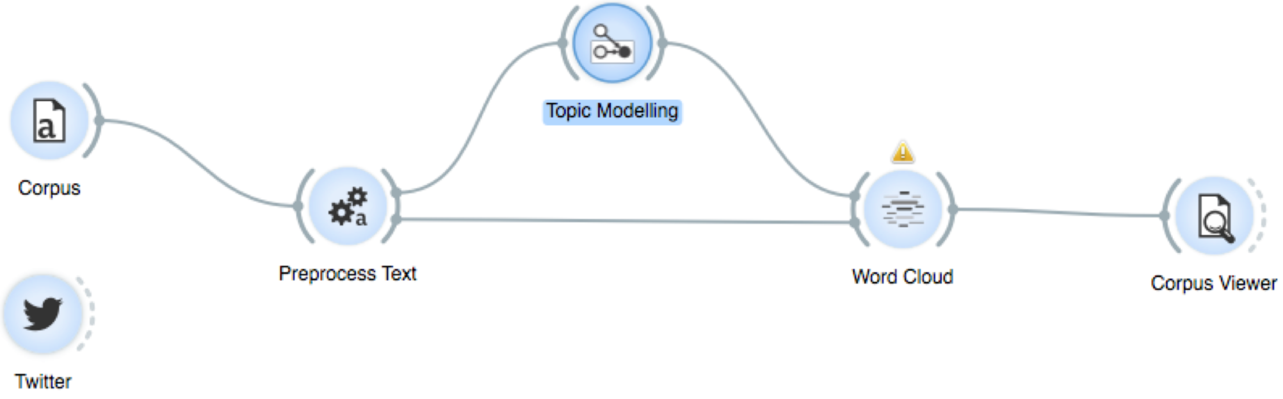
44 document(s), 4 text features(s), 1 other feature(s).
Classification; discrete class with 2 values.

Used text features

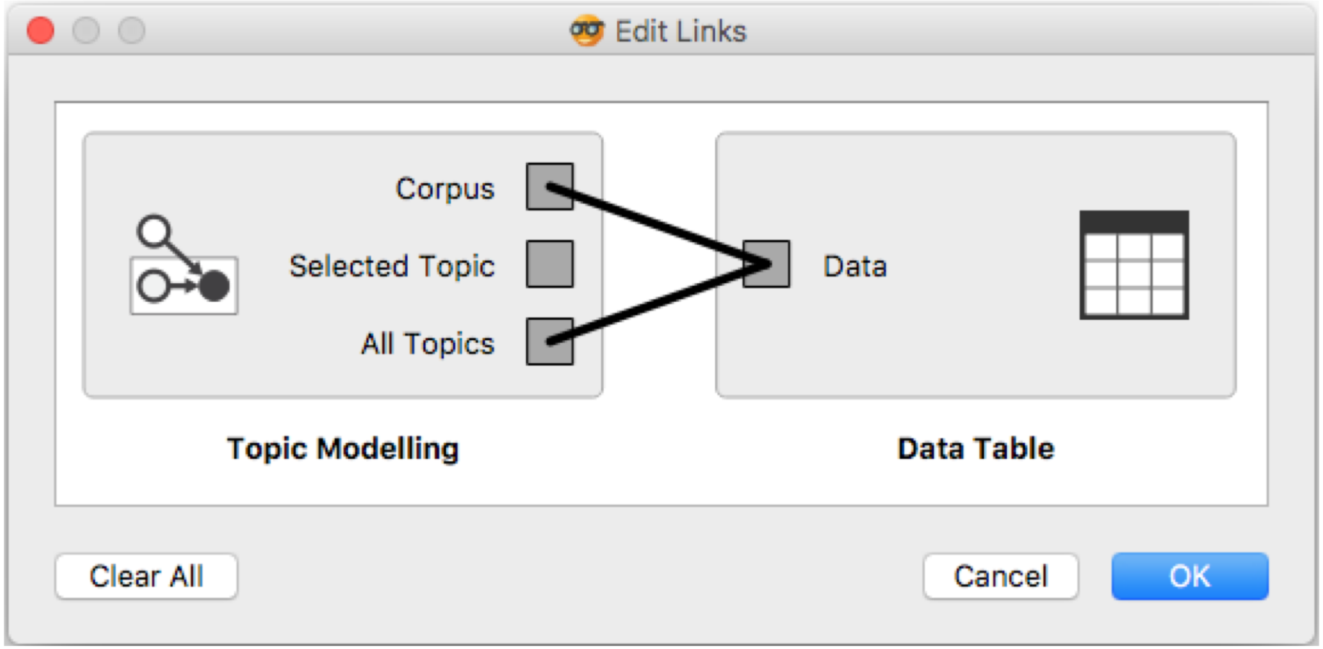
- S Content

Ignored text features

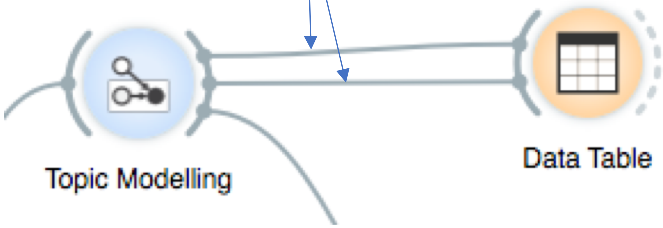
- S Title
- S Abstract
- S ATU Numerical



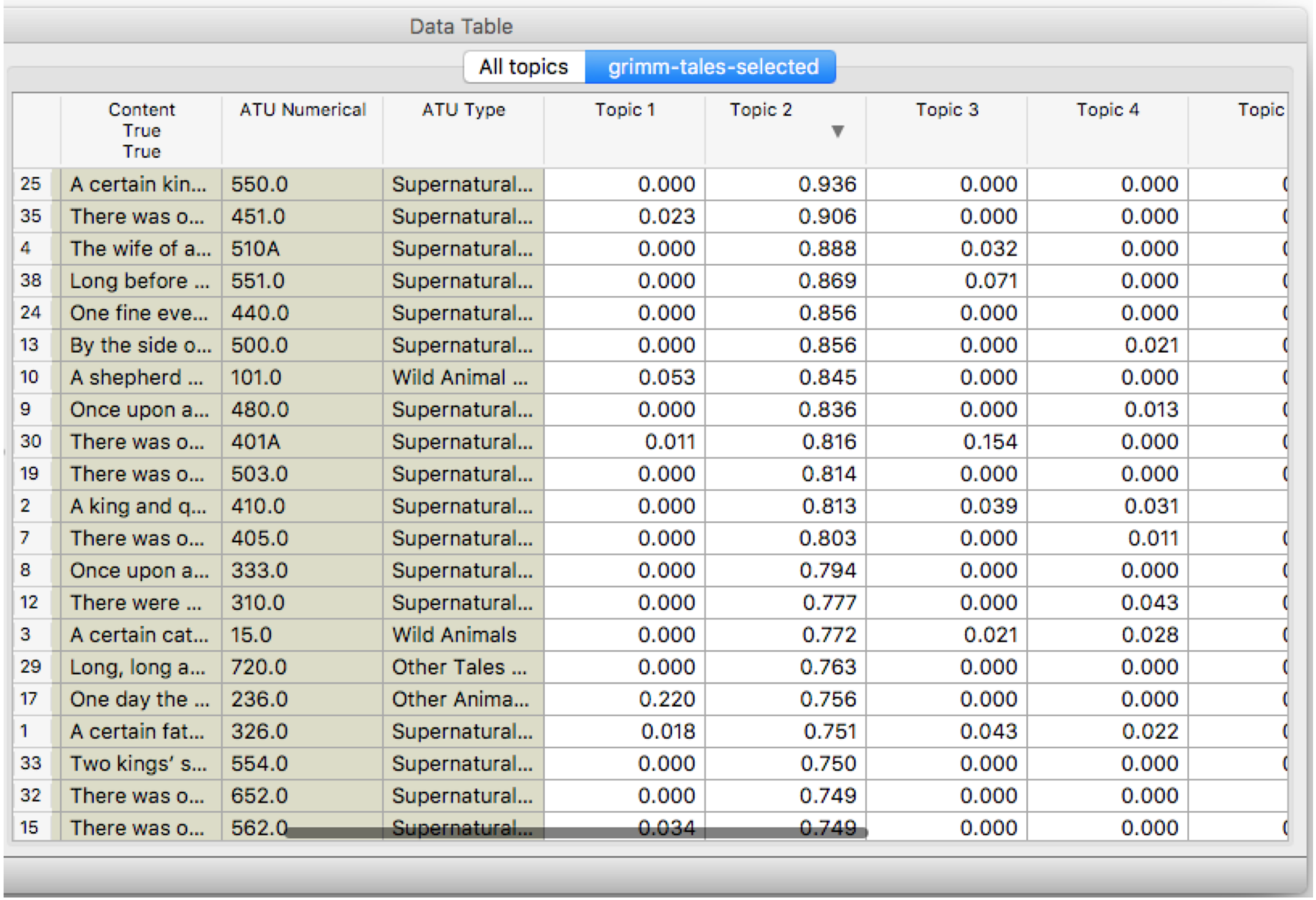
Visualizing by table



Double click

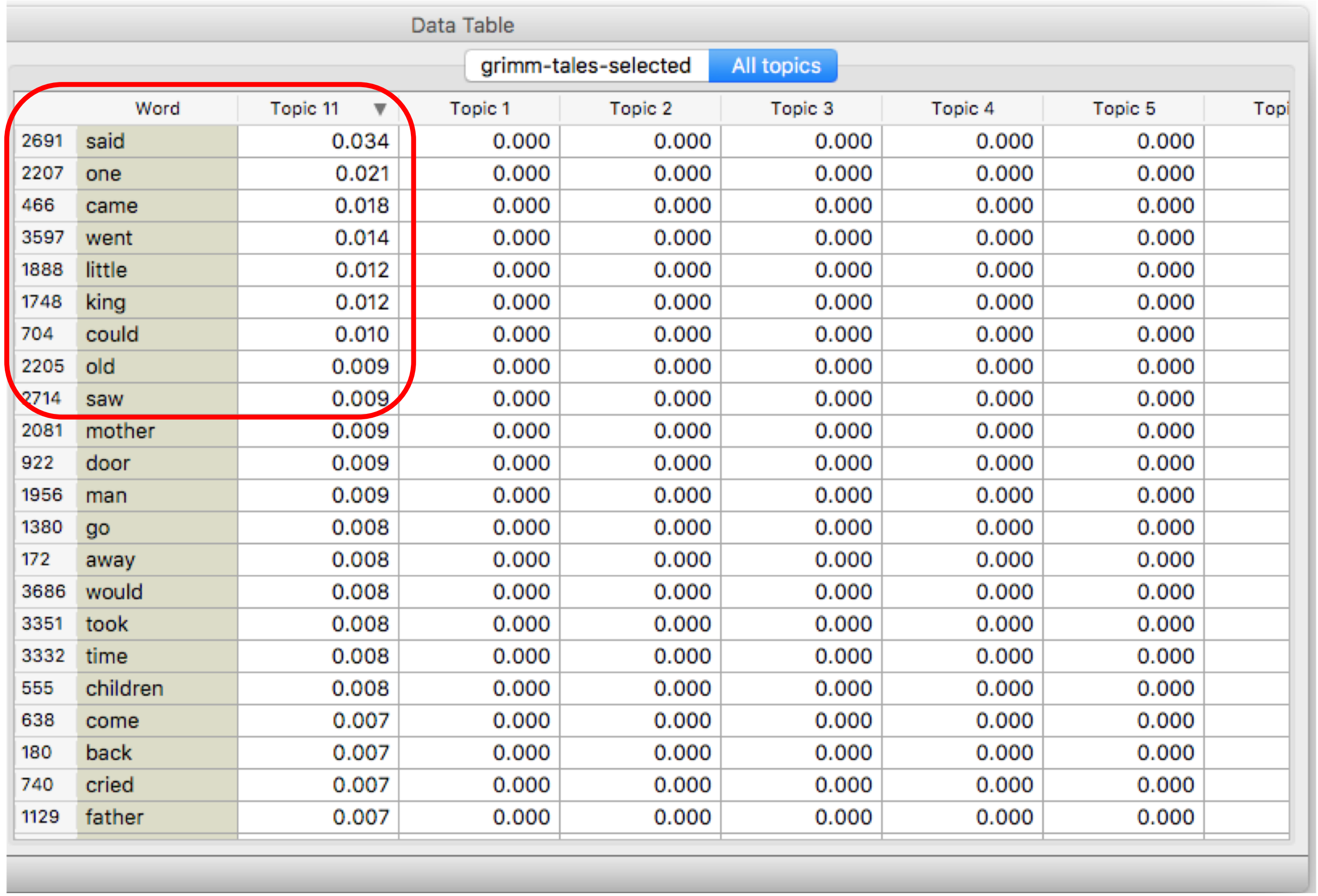


"Topic 2" is heavily affecting results maybe its better to increase number of topics



	Content True True	ATU Numerical	ATU Type	Topic 1	Topic 2 ▼	Topic 3	Topic 4	Topic
25	A certain kin...	550.0	Supernatural...	0.000	0.936	0.000	0.000	
35	There was o...	451.0	Supernatural...	0.023	0.906	0.000	0.000	
4	The wife of a...	510A	Supernatural...	0.000	0.888	0.032	0.000	
38	Long before ...	551.0	Supernatural...	0.000	0.869	0.071	0.000	
24	One fine eve...	440.0	Supernatural...	0.000	0.856	0.000	0.000	
13	By the side o...	500.0	Supernatural...	0.000	0.856	0.000	0.021	
10	A shepherd ...	101.0	Wild Animal ...	0.053	0.845	0.000	0.000	
9	Once upon a...	480.0	Supernatural...	0.000	0.836	0.000	0.013	
30	There was o...	401A	Supernatural...	0.011	0.816	0.154	0.000	
19	There was o...	503.0	Supernatural...	0.000	0.814	0.000	0.000	
2	A king and q...	410.0	Supernatural...	0.000	0.813	0.039	0.031	
7	There was o...	405.0	Supernatural...	0.000	0.803	0.000	0.011	
8	Once upon a...	333.0	Supernatural...	0.000	0.794	0.000	0.000	
12	There were ...	310.0	Supernatural...	0.000	0.777	0.000	0.043	
3	A certain cat...	15.0	Wild Animals	0.000	0.772	0.021	0.028	
29	Long, long a...	720.0	Other Tales ...	0.000	0.763	0.000	0.000	
17	One day the ...	236.0	Other Anima...	0.220	0.756	0.000	0.000	
1	A certain fat...	326.0	Supernatural...	0.018	0.751	0.043	0.022	
33	Two kings' s...	554.0	Supernatural...	0.000	0.750	0.000	0.000	
32	There was o...	652.0	Supernatural...	0.000	0.749	0.000	0.000	
15	There was o...	562.0	Supernatural...	0.034	0.749	0.000	0.000	

If we increase topic number to 20, now "Topic 11" is having high impact
Maybe there are too general words included there?

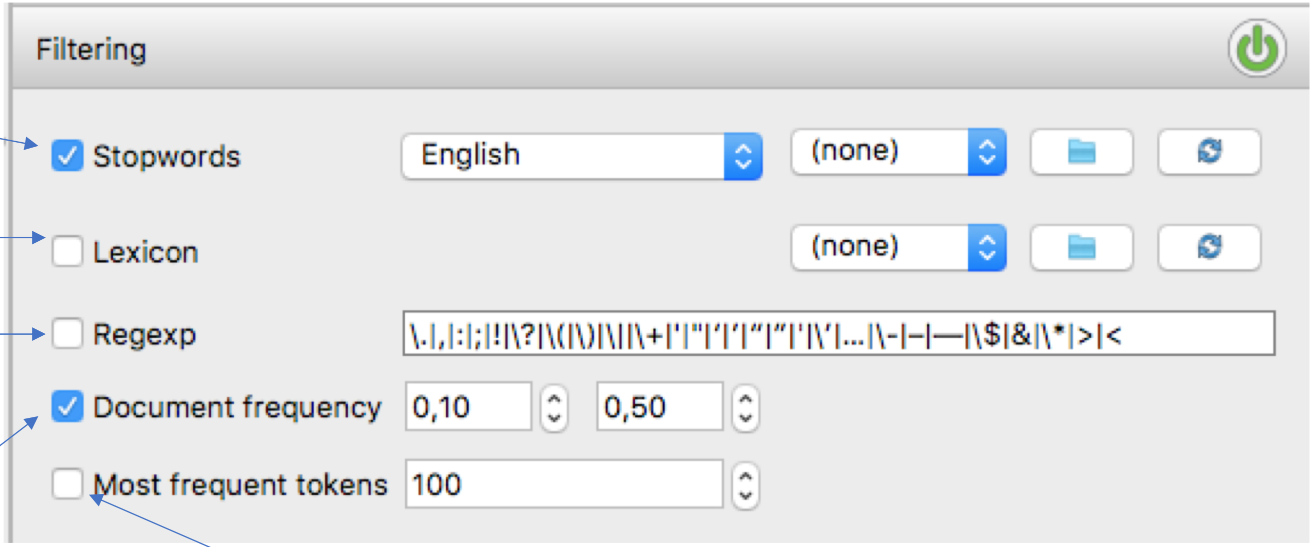


	Word	Topic 11 ▼	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topi
2691	said	0.034	0.000	0.000	0.000	0.000	0.000	
2207	one	0.021	0.000	0.000	0.000	0.000	0.000	
466	came	0.018	0.000	0.000	0.000	0.000	0.000	
3597	went	0.014	0.000	0.000	0.000	0.000	0.000	
1888	little	0.012	0.000	0.000	0.000	0.000	0.000	
1748	king	0.012	0.000	0.000	0.000	0.000	0.000	
704	could	0.010	0.000	0.000	0.000	0.000	0.000	
2205	old	0.009	0.000	0.000	0.000	0.000	0.000	
2714	saw	0.009	0.000	0.000	0.000	0.000	0.000	
2081	mother	0.009	0.000	0.000	0.000	0.000	0.000	
922	door	0.009	0.000	0.000	0.000	0.000	0.000	
1956	man	0.009	0.000	0.000	0.000	0.000	0.000	
1380	go	0.008	0.000	0.000	0.000	0.000	0.000	
172	away	0.008	0.000	0.000	0.000	0.000	0.000	
3686	would	0.008	0.000	0.000	0.000	0.000	0.000	
3351	took	0.008	0.000	0.000	0.000	0.000	0.000	
3332	time	0.008	0.000	0.000	0.000	0.000	0.000	
555	children	0.008	0.000	0.000	0.000	0.000	0.000	
638	come	0.007	0.000	0.000	0.000	0.000	0.000	
180	back	0.007	0.000	0.000	0.000	0.000	0.000	
740	cried	0.007	0.000	0.000	0.000	0.000	0.000	
1129	father	0.007	0.000	0.000	0.000	0.000	0.000	

Words list can be uploaded and only words that are in the list - removed

Words list can be uploaded and only words that are in the list - removed

Removes what is located with Regex



Filtering

☑ Stopwords English (none)

□ Lexicon (none)

□ Regexp `\.|,|:|;|!|\?|\(|\)|\||\+|'|"|‘|’|“|”|'|\'|…|\-|–|—|\$|&|\*|>|<`

☑ Document frequency 0,10 0,50

□ Most frequent tokens 100

If integers are provided as parameters, it keeps only tokens that appear in the specified number of documents. **E.g. DF = (3, 5). keeps only tokens that appear in 3 or more and 5 or less documents**. If floats are provided, it keeps only tokens that appear in the specified percentage of documents. E.g. **DF = (0.3, 0.5) keeps only tokens that appear in 30% to 50% of documents**

Keeps only specified number of most frequent tokens (words/phrases)