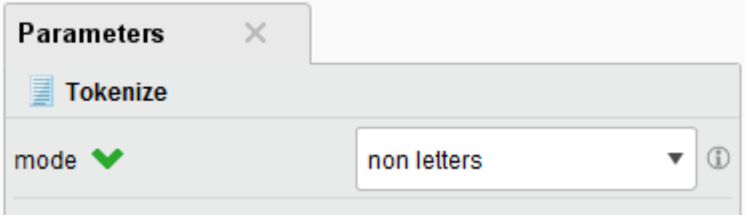
Text Mining

Exercise 7

Text Preprocessing

- Tokenisation
  - Break text into single words or n-grams
  - “example text”
    - (“example”, “text”)
    - (“exam”, “xamp”, “ampl”, “mple”, “ple ”, “le t”, “e te”, “ tex”, “text”)
- Stopword Removal
  - Remove frequent words that may confuse your algorithm
  - “this is an example” -> “example”
- Stemming
  - Finding the root/stem of a word helps matching similar words
  - “user”, “users”, “used”, “using” -> “use”

Operators: Tokenize

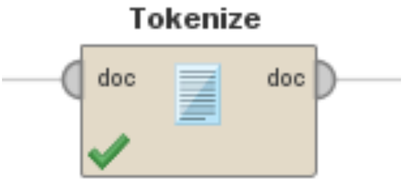


Parameters

Tokenize

mode ✓ non letters

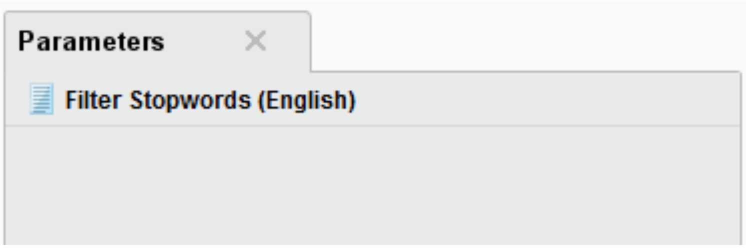
- Input Port
  - Document
- Output Port
  - Tokenised Document
- Parameters
  - Mode (how to create tokens)



- Splits a document into tokens

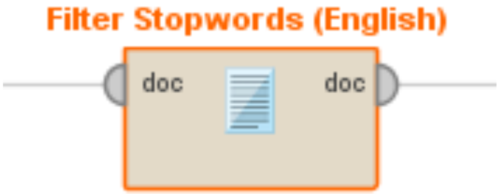
Operators: Filter Stopwords (English)

- Input Port
  - Tokenised Document
- Output Port
  - Tokenised Document
- Parameters
  - None



Parameters ✕

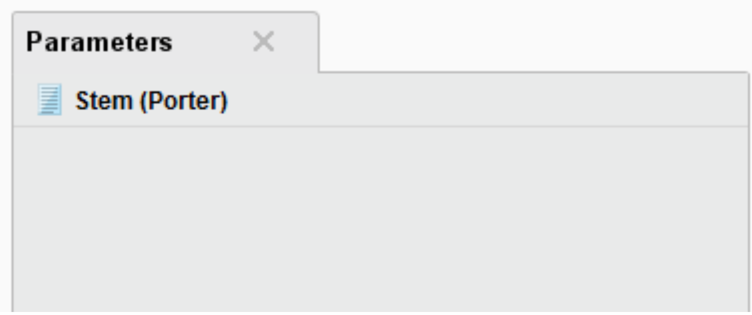
Filter Stopwords (English)



- Removes stopwords
- Different operators for different languages

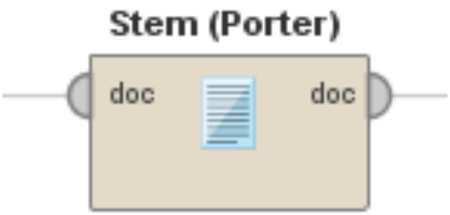
Operators: Stem (Porter)

- Input Port
  - Tokenised Document
- Output Port
  - Tokenised Document
- Parameters
  - None



Parameters

Stem (Porter)



- Replaces tokens with their stems
- Different operators for different stemming methods

Feature Generation from Text

- Documents are treated as bags of words (tokens)
  - Each token becomes a feature
  - The order of tokens is ignored
- Different techniques to determine feature values (feature vector creation)
  - Binary Term Occurrence: 1 if the token is present, 0 otherwise
  - Term Occurrence: Absolute frequency of the token, i.e., 5
  - Term Frequency: Relative frequency of the token, i.e., 5%
  - Term Frequency – Inverse Document Frequency:
    - More weight if the token is rare
    - Less weight if the token is frequent

Feature Generation Examples – Binary Term Occurrences

- Sample document set:
  - d1 = “Saturn is the gas planet with rings.”
  - d2 = “Jupiter is the largest gas planet.”
  - d3 = “Saturn is the Roman god of sowing.”
- Documents as vectors:

	saturn	is	the	gas	planet	with	rings	jupiter	largest	roman	god	of	sowing
D1	1	1	1	1	1	1	1	0	0	0	0	0	0
D2	0	1	1	1	1	0	0	1	1	0	0	0	0
D3	1	1	1	0	0	0	0	0	0	1	1	1	1

Feature Generation Examples –Term Frequency

- Sample document set:
  - d1 = “Saturn is the gas planet with rings.”
  - d2 = “Jupiter is the largest gas planet.”
  - d3 = “Saturn is the Roman god of sowing.”
- Documents as vectors:

	saturn	is	the	gas	planet	with	rings	jupiter	largest	roman	god	of	sowing
D1	1/7	1/7	1/7	1/7	1/7	1/7	1/7	0	0	0	0	0	0
D2	0	1/6	1/6	1/6	1/6	0	0	1/6	1/6	0	0	0	0
D3	1/7	1/7	1/7	0	0	0	0	0	0	1/7	/71	1/7	1/7

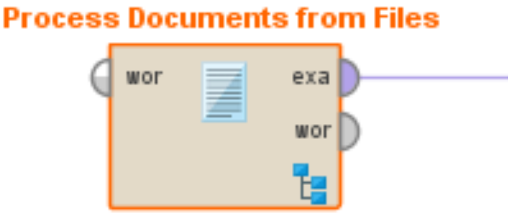
Feature Generation Examples – TF-IDF

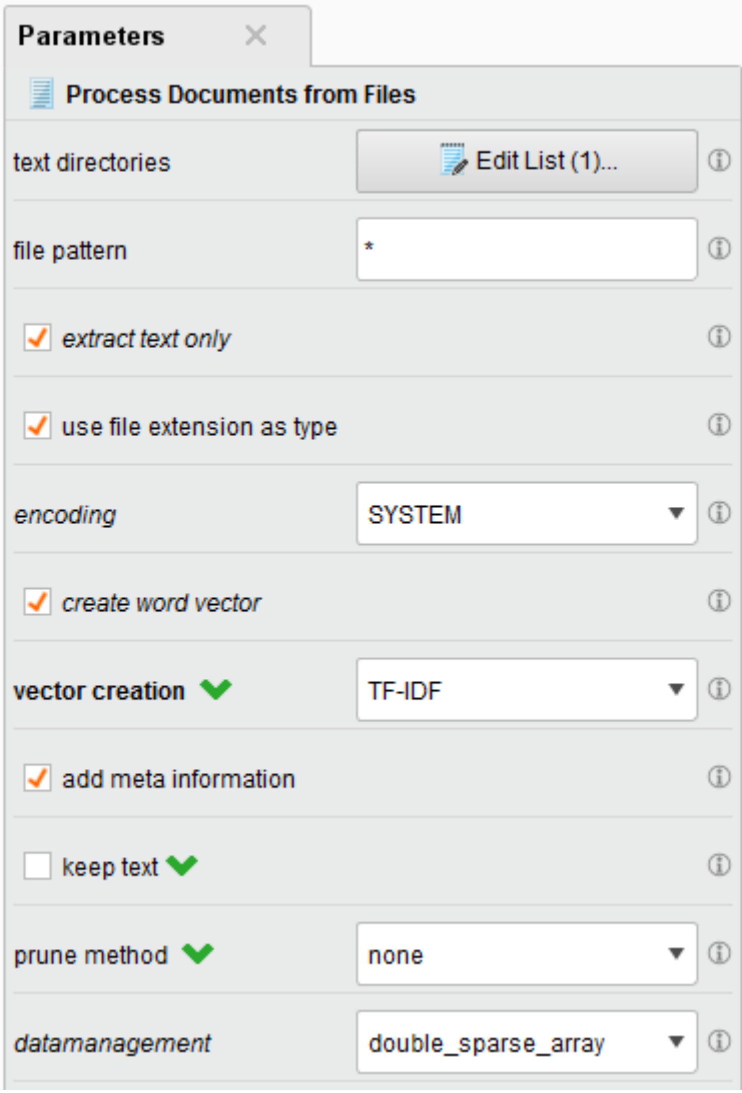
- Sample document set:
  - d1 = “Saturn is the gas planet with rings.”
  - d2 = “Jupiter is the largest gas planet.”
  - d3 = “Saturn is the Roman god of sowing.”
- Documents as vectors:

	saturn	is	the	gas	planet	with	rings	jupiter	largest	roman	god	of	sowing
D1	0.03	0	0	0.03	0.03	0.07	0.07	0	0	0	0	0	0
D2	0	0	0	0.03	0.03	0	0	0.08	0.08	0	0	0	0
D3	0.03	0	0	0	0	0	0	0	0	0.07	0.07	0.07	0.07

Operators: Process Documents from Files

- Input Port
  - Word Vector (optional)
- Output Ports
  - Example Set (Vectorised Documents)
  - Word Vector
- Parameters
  - Directories: Which files to load & which label to assign
  - Vector creation method
  - Pruning (next slide)





text directories: Edit List (1)...

file pattern: *

☑ extract text only

☑ use file extension as type

encoding: SYSTEM

☑ create word vector

vector creation: TF-IDF

☑ add meta information

□ keep text

prune method: none

datamanagement: double_sparse_array

Feature Selection

- High dimensional data!
- Not all features help!

- Pruning: Remove too frequent or too infrequent tokens
  - Percentual: ignore words that appear in less / more than a given percentage of all documents
  - Absolute: ignore words that appear in less / more than a given number of documents
  - By Rank: ignore a given percentage of the most frequent / infrequent words

Similarity Measures for Documents: Jaccard Coefficient

- Jaccard Coefficient:
  - For asymmetric binary attributes: the 1 state is more important than the 0 state

$$Jaccard(x_i, x_j) = \frac{M_{11}}{M_{01} + M_{10} + M_{11}}$$

	saturn	is	the	gas	planet	with	rings	jupiter	largest	roman	god	of	sowing
D1	1	1	1	1	1	1	1	0	0	0	0	0	0
D2	0	1	1	1	1	0	0	1	1	0	0	0	0
D3	1	1	1	0	0	0	0	0	0	1	1	1	1

- With stopwords

$$Jaccard(D1, D2) = \frac{4}{2+3+4} = 0.44$$

$$Jaccard(D1, D3) = \frac{3}{4+4+3} = 0.27$$

$$Jaccard(D2, D3) = \frac{2}{5+4+2} = 0.18$$

Without stopwords

$$Jaccard(D1, D2) = \frac{2}{2+2+2} = 0.33$$

$$Jaccard(D1, D3) = \frac{1}{3+3+1} = 0.14$$

$$Jaccard(D2, D3) = \frac{0}{4+4+0} = 0.00$$

Similarity Measures for Documents: Cosine Similarity

- Cosine Similarity
  - Dot product only considers combinations that are both non-zero
  - Normalised by length of both vectors

$$\cos(d_1, d_2) = \frac{d_1 \cdot d_2}{|d_1|\ |d_2|}$$

	saturn	is	the	gas	planet	with	rings	jupiter	largest	roman	god	of	sowing
D1	0.03	0	0	0.03	0.03	0.07	0.07	0	0	0	0	0	0
D2	0	0	0	0.03	0.03	0	0	0.08	0.08	0	0	0	0
D3	0.03	0	0	0	0	0	0	0	0	0.07	0.07	0.07	0.07

- With stopwords

$Cosine(D1, D2) = 0.13$

$Cosine(D1, D3) = 0.05$

$Cosine(D2, D3) = 0.00$

Without stopwords

$Cosine(D1, D2) = 0.17$

$Cosine(D1, D3) = 0.08$

$Cosine(D2, D3) = 0.00$

Today's Datasets

- Corpus 4-docs:
  - Doc1: “David Cameron Joins Talks On Euro Crisis”
  - Doc2: “Real Madrid Slips Into First With a Hat Trick by Ronaldo”
  - Doc3: “An Occupation for the 99 Per Cent” (Occupy Wall Street)
  - Doc4: “Málaga vs. Real Madrid Barcelona vs. Sevilla”
- Corpus 30-docs (newsgroups[1]):
  - sci.space
  - soc.religion.christian
  - talk.politics.guns
- Corpus 300-docus:
  - misc.forsale
  - rec.sport.baseball
  - rec.sport.hockey
- Job Postings:
  - Category + Posting Text

[1] https://en.wikipedia.org/wiki/Usenet_newsgroup