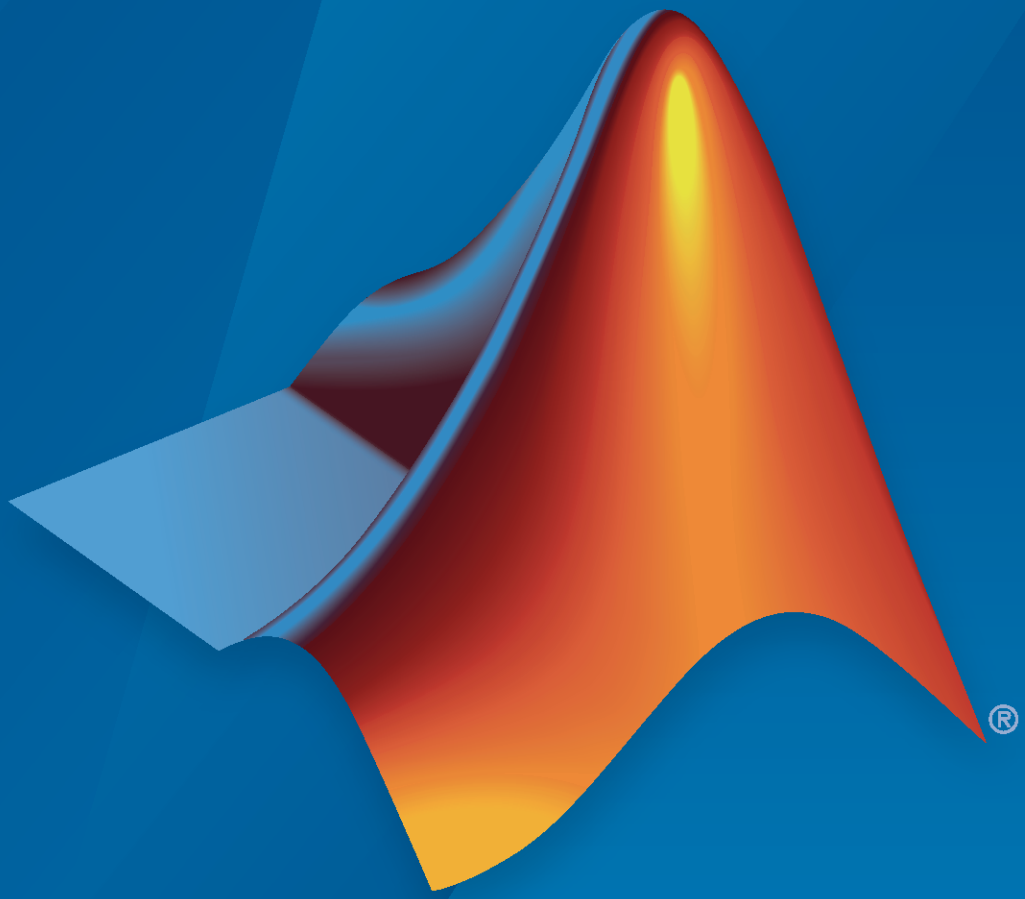
Text Analytics Toolbox™

Getting Started Guide



MATLAB®

R2021a

MathWorks®

How to Contact MathWorks

Latest news: `www.mathworks.com`

Sales and services: `www.mathworks.com/sales_and_services`

User community: `www.mathworks.com/matlabcentral`

Technical support: `www.mathworks.com/support/contact_us`

Phone: 508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

Text Analytics Toolbox™ Getting Started Guide

© COPYRIGHT 2017–2021 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See `www.mathworks.com/trademarks` for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see `www.mathworks.com/patents` for more information.

Revision History

September 2017	Online Only	New for Version 1.0
March 2018	Online Only	Revised for Version 1.1 (Release 2018a)
September 2018	Online Only	Revised for Version 1.2 (Release 2018b)
March 2019	Online Only	Revised for Version 1.3 (Release 2019a)
September 2019	Online Only	Revised for Version 1.4 (Release 2019b)
March 2020	Online Only	Revised for Version 1.5 (Release 2020a)
September 2020	Online Only	Revised for Version 1.6 (Release 2020b)
March 2021	Online Only	Revised for Version 1.7 (Release 2021a)

Contents

1 Getting Started

Text Analytics Toolbox Product Description **1-2**
Key Features **1-2**

Try Text Analytics in 10 Lines of Code **1-3**

Import Text Data into MATLAB **1-5**

Create Simple Preprocessing Function **1-6**

Get Started with Topic Modeling **1-9**

Getting Started

- “Text Analytics Toolbox Product Description” on page 1-2
- “Try Text Analytics in 10 Lines of Code” on page 1-3
- “Import Text Data into MATLAB” on page 1-5
- “Create Simple Preprocessing Function” on page 1-6
- “Get Started with Topic Modeling” on page 1-9

Text Analytics Toolbox Product Description

Analyze and model text data

Text Analytics Toolbox provides algorithms and visualizations for preprocessing, analyzing, and modeling text data. Models created with the toolbox can be used in applications such as sentiment analysis, predictive maintenance, and topic modeling.

Text Analytics Toolbox includes tools for processing raw text from sources such as equipment logs, news feeds, surveys, operator reports, and social media. You can extract text from popular file formats, preprocess raw text, extract individual words, convert text into numerical representations, and build statistical models.

Using machine learning techniques such as LSA, LDA, and word embeddings, you can find clusters and create features from high-dimensional text datasets. Features created with Text Analytics Toolbox can be combined with features from other data sources to build machine learning models that take advantage of textual, numeric, and other types of data.

Key Features

- Text preprocessing and normalization
- Machine learning algorithms, including latent Dirichlet allocation (LDA) and latent semantic analysis (LSA)
- Word-embedding training, and pretrained model import from word2vec, FastText, and GloVe
- Word cloud and text scatter plots
- Document import from PDF and Microsoft® Word files
- TF-IDF and word frequency statistics

Try Text Analytics in 10 Lines of Code

This example shows how to use text analytics to classify text data using only 10 lines of MATLAB® code. Try the example to see how simple it is to get started with text analytics in MATLAB.

You can create a simple classification model which uses word frequency counts as predictors. This example trains a classification model to predict the event type of factory reports using text descriptions.

Create Model

The main steps of creating a model are:

1. Import – import the text data into MATLAB.
2. Preprocess – preprocess the text for word analysis.
3. Convert – convert the text to numeric data.
4. Train – train a classification model.

Import the example text data and labels, tokenize the text, convert it to numeric data using a bag-of-words model, and train a supervised SVM classifier.

```
data = readtable('factoryReports.csv','TextType','String'); % Read data
labels = categorical(data.Category);                        % Read labels

documents = tokenizedDocument(data.Description);            % Preprocess text

bag = bagOfWords(documents);                                % Count words
XTrain = bag.Counts;                                        % Convert to numeric data

mdl = fitcecoc(XTrain,labels,'Learners','linear');          % Train classifier
```

Predict Using New Data

The steps for prediction are similar to those for training. To predict using new data, preprocess the text data and convert it to numeric using the same steps used for training. Then, predict the label using the trained model.

Predict the label for the text `"Coolant is pooling underneath sorter."`.

```
str = "Coolant is pooling underneath sorter.";              % Import text
documentsNew = tokenizedDocument(str);                      % Preprocess text
XTest = encode(bag,documentsNew);                           % Convert to numeric
label = predict(mdl,XTest)                                  % Predict label
```

```
label = categorical
     Leak
```

For an example showing a more detailed workflow, see “Create Simple Text Model for Classification”.

For next steps in text analytics, you can try improving the model accuracy by preprocessing the data and visualize the text data using word clouds. For examples, see “Prepare Text Data for Analysis” and “Visualize Text Data Using Word Clouds”.

See Also

`bagOfWords` | `encode` | `tokenizedDocument`

Related Examples

- “Analyze Text Data Using Topic Models”
- “Analyze Text Data Using Multiword Phrases”
- “Analyze Text Data Containing Emojis”
- “Train a Sentiment Classifier”
- “Classify Text Data Using Deep Learning”
- “Generate Text Using Deep Learning” (Deep Learning Toolbox)

Import Text Data into MATLAB

Use the following functions to import text data into MATLAB®. Usually, the easiest way to import text data into MATLAB is to use the `extractFileText` function. For example, to import text from a text file, use:

```
str = extractFileText('sonnets.txt');
```

This table outlines which function to use for different file types. For more information, see “Extract Text Data from Files”.

Source	Function	Example
Text, Microsoft Word, PDF, and HTML files	`extractFileText`	`str = extractFileText('sonnets.txt');`
CSV and Microsoft Excel® files	`readtable`	`tbl = readtable('weatherReports.csv','TextType','string');`
HTML code	`extractHTMLText`	`str = extractHTMLText('<html><body><h1>THE SONNETS</h1><p>by William Shakespeare</p></body></html>');`
PDF forms	`readPDFFormData`	`data = readPDFFormData('weatherReportForm1.pdf');`

See Also

`extractFileText` | `extractHTMLText` | `readPDFFormData` | `readtable`

More About

- “Try Text Analytics in 10 Lines of Code” on page 1-3
- “Create Simple Preprocessing Function” on page 1-6
- “Get Started with Topic Modeling” on page 1-9
- “Visualize Text Data Using Word Clouds”
- “Text Analytics Glossary”

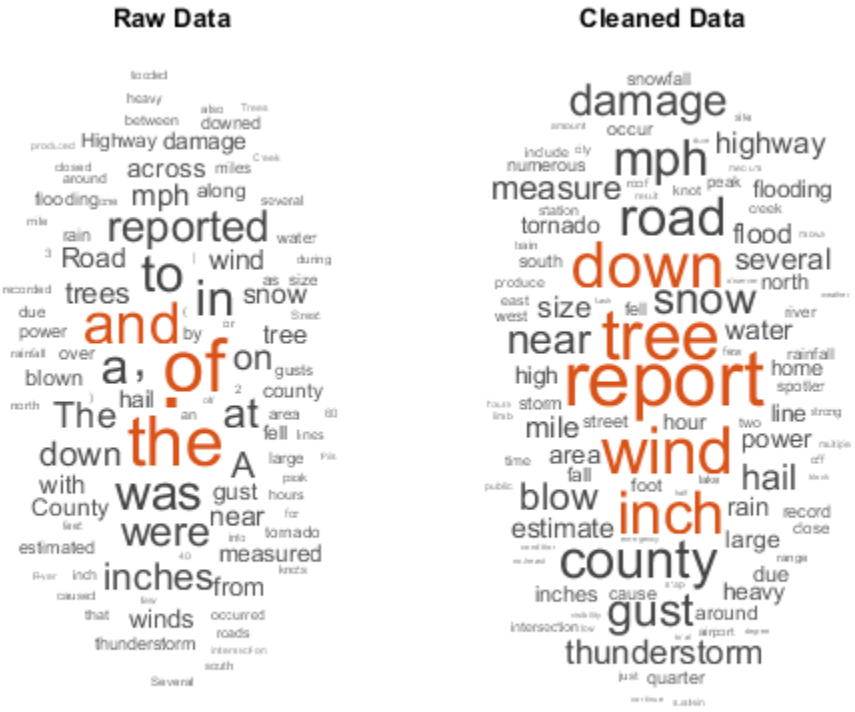
Create Simple Preprocessing Function

This example shows how to create a function which cleans and preprocesses text data for analysis.

Text data can be large and can contain lots of noise which negatively affects statistical analysis. For example, text data can contain the following:

- Variations in case, for example "new" and "New"
- Variations in word forms, for example "walk" and "walking"
- Words which add noise, for example "stop words" such as "the" and "of"
- Punctuation and special characters
- HTML and XML tags

These word clouds illustrate word frequency analysis applied to some raw text data from weather reports, and a preprocessed version of the same text data.



It can be useful to create a preprocessing function, so you can prepare different collections of text data in the same way. For example, when training a model, you can use a function so that you can preprocess new data using the same steps as the training data.

The function `preprocessTextData`, listed at the end of the example, performs the following steps:

1. Tokenize the text using `tokenizedDocument`.
2. Lemmatize the words using `normalizeWords`.

3. Erase punctuation using `erasePunctuation`.
4. Remove a list of stop words (such as "and", "of", and "the") using `removeStopWords`.
5. Remove words with 2 or fewer characters using `removeShortWords`.
6. Remove words with 15 or more characters using `removeLongWords`.

To use the function, simply input your text data into `preprocessTextData`.

```
textData = [
    "A large tree is downed and blocking traffic outside Apple Hill."
    "There is lots of damage to many car windshields in the parking lot."];
documents = preprocessTextData(textData)
```

```
documents =
  2x1 tokenizedDocument:

    8 tokens: large tree down block traffic outside apple hill
    7 tokens: lot damage many car windshield parking lot
```

Preprocessing Function

```
function documents = preprocessTextData(textData)

% Tokenize the text.
documents = tokenizedDocument(textData);

% Lemmatize the words. To improve lemmatization, first use
% addPartOfSpeechDetails.
documents = addPartOfSpeechDetails(documents);
documents = normalizeWords(documents,'Style','lemma');

% Erase punctuation.
documents = erasePunctuation(documents);

% Remove a list of stop words.
documents = removeStopWords(documents);

% Remove words with 2 or fewer characters, and words with 15 or more
% characters.
documents = removeShortWords(documents,2);
documents = removeLongWords(documents,15);

end
```

For an example showing a more detailed workflow, see “Prepare Text Data for Analysis”.

For next steps in text analytics, you can try creating a classification model or analyze the data using topic models. For examples, see “Create Simple Text Model for Classification” and “Analyze Text Data Using Topic Models”.

See Also

`addPartOfSpeechDetails` | `erasePunctuation` | `normalizeWords` | `removeLongWords` | `removeShortWords` | `removeStopWords` | `tokenizedDocument`

More About

- “Try Text Analytics in 10 Lines of Code” on page 1-3
- “Import Text Data into MATLAB” on page 1-5
- “Get Started with Topic Modeling” on page 1-9
- “Visualize Text Data Using Word Clouds”
- “Text Analytics Glossary”

Get Started with Topic Modeling

This example shows how to fit a topic model to text data and visualize the topics.

A Latent Dirichlet Allocation (LDA) model is a topic model which discovers underlying topics in a collection of documents. Topics, characterized by distributions of words, correspond to groups of commonly co-occurring words. LDA is an unsupervised topic model which means that it does not require labeled data.

Load and Extract Text Data

Load the example data. The file `factoryReports.csv` contains factory reports, including a text description and categorical labels for each event.

Import the data using the `readtable` function and extract the text data from the `Description` column.

```
filename = "factoryReports.csv";
data = readtable(filename,'TextType','string');
textData = data.Description;
```

Prepare Text Data for Analysis

Tokenize and preprocess the text data and create a bag-of-words model.

Tokenize the text.

```
documents = tokenizedDocument(textData);
```

To improve the model fit, remove the punctuation and the stop words (words like "and", "of", and "the") from the documents.

```
documents = removeStopWords(documents);
documents = erasePunctuation(documents);
```

Create a bag-of-words model.

```
bag = bagOfWords(documents);
```

Fit LDA Model

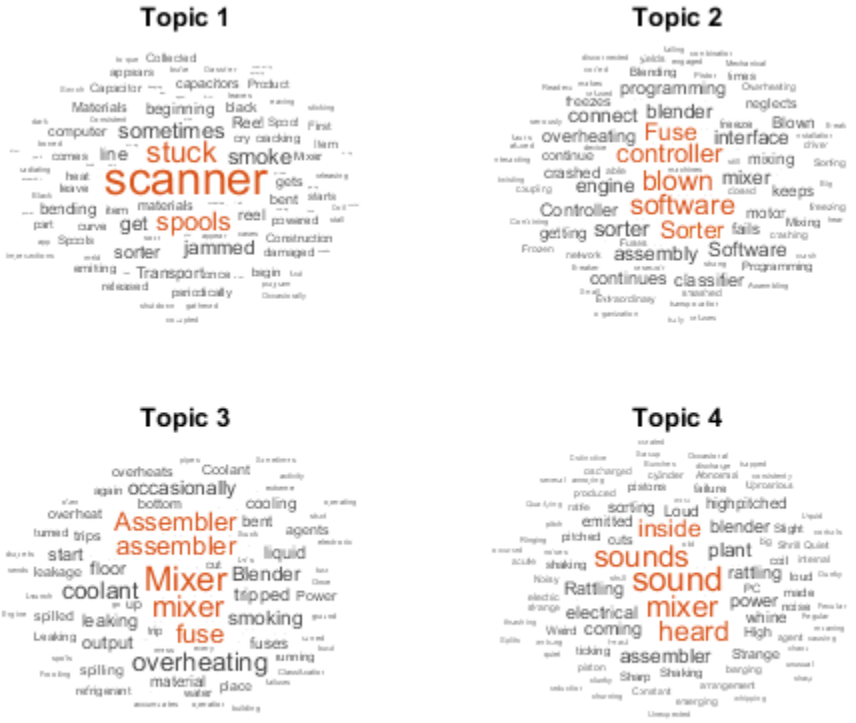
Fit an LDA model with seven topics using the `fitlda` function. To suppress the verbose output, set the `'Verbose'` option to `0`.

```
numTopics = 7;
mdl = fitlda(bag,numTopics,'Verbose',0);
```

Visualize Topics

Visualize the first four topics using word clouds.

```
figure
for topicIdx = 1:4
    subplot(2,2,topicIdx)
    wordcloud(mdl,topicIdx);
    title("Topic " + topicIdx)
end
```



For next steps in text analytics, you can try improving the model fit by using different preprocessing steps and visualizing the topic mixtures. For an example, see “Analyze Text Data Using Topic Models”.

See Also

`bagOfWords` | `eraseStopWords` | `fitlda` | `removeStopWords` | `tokenizedDocument` | `wordcloud`

More About

- “Try Text Analytics in 10 Lines of Code” on page 1-3
- “Import Text Data into MATLAB” on page 1-5
- “Create Simple Preprocessing Function” on page 1-6
- “Visualize Text Data Using Word Clouds”
- “Text Analytics Glossary”