以下资料来自英文网站

<https://www.mathworks.com/help/textanalytics/examples.html>

是机器翻译的结果，请对照英文资料，学习并运行所有代码，修改文字使之通顺流畅。

（使用Word的‘审阅’修订功能进行修改，保留所有修改痕迹）

# 文本分析工具箱入门

## 创建用于分类的简单文本模型

此示例演示如何使用一个单词袋模型训练一个简单的文本分类器上的单词频率计数。

您可以创建一个简单的分类模型, 使用 word 频率计数作为预测因子。本示例使用文本描述训练一个简单的分类模型来预测天气报告的事件类型。

若要重现此示例的结果, 请将rng设置为'default'.

rng('default')

### 加载和提取文本数据

加载示例数据。文件weatherReports.csv包含天气报告, 包括每个事件的文本说明和分类标签。

filename = "weatherReports.csv";

data = readtable(filename,'TextType','string');

head(data)

ans=8×16 table

Time event\_id state event\_type damage\_property damage\_crops begin\_lat begin\_lon end\_lat end\_lon event\_narrative storm\_duration begin\_day end\_day year end\_timestamp

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

22-Jul-2016 16:10:00 6.4433e+05 "MISSISSIPPI" "Thunderstorm Wind" "" "0.00K" 34.14 -88.63 34.122 -88.626 "Large tree down between Plantersville and Nettleton." 00:05:00 22 22 2016 22-Jul-0016 16:15:00

15-Jul-2016 17:15:00 6.5182e+05 "SOUTH CAROLINA" "Heavy Rain" "2.00K" "0.00K" 34.94 -81.03 34.94 -81.03 "One to two feet of deep standing water developed on a street on the Winthrop University campus after more than an inch of rain fell in less than an hour. One vehicle was stalled in the water." 00:00:00 15 15 2016 15-Jul-0016 17:15:00

15-Jul-2016 17:25:00 6.5183e+05 "SOUTH CAROLINA" "Thunderstorm Wind" "0.00K" "0.00K" 35.01 -80.93 35.01 -80.93 "NWS Columbia relayed a report of trees blown down along Tom Hall St." 00:00:00 15 15 2016 15-Jul-0016 17:25:00

16-Jul-2016 12:46:00 6.5183e+05 "NORTH CAROLINA" "Thunderstorm Wind" "0.00K" "0.00K" 35.64 -82.14 35.64 -82.14 "Media reported two trees blown down along I-40 in the Old Fort area." 00:00:00 16 16 2016 16-Jul-0016 12:46:00

15-Jul-2016 14:28:00 6.4332e+05 "MISSOURI" "Hail" "" "" 36.45 -89.97 36.45 -89.97 "" 00:07:00 15 15 2016 15-Jul-0016 14:35:00

15-Jul-2016 16:31:00 6.4332e+05 "ARKANSAS" "Thunderstorm Wind" "" "0.00K" 35.85 -90.1 35.838 -90.087 "A few tree limbs greater than 6 inches down on HWY 18 in Roseland." 00:09:00 15 15 2016 15-Jul-0016 16:40:00

15-Jul-2016 16:03:00 6.4343e+05 "TENNESSEE" "Thunderstorm Wind" "20.00K" "0.00K" 35.056 -89.937 35.05 -89.904 "Awning blown off a building on Lamar Avenue. Multiple trees down near the intersection of Winchester and Perkins." 00:07:00 15 15 2016 15-Jul-0016 16:10:00

15-Jul-2016 17:27:00 6.4344e+05 "TENNESSEE" "Hail" "" "" 35.385 -89.78 35.385 -89.78 "Quarter size hail near Rosemark." 00:05:00 15 15 2016 15-Jul-0016 17:32:00

删除具有空报表的行。

idx = strlength(data.event\_narrative) == 0;

data(idx,:) = [];

将表的event\_type列中的标签转换为分类, 并使用直方图查看数据中类的分布。

data.event\_type = categorical(data.event\_type);

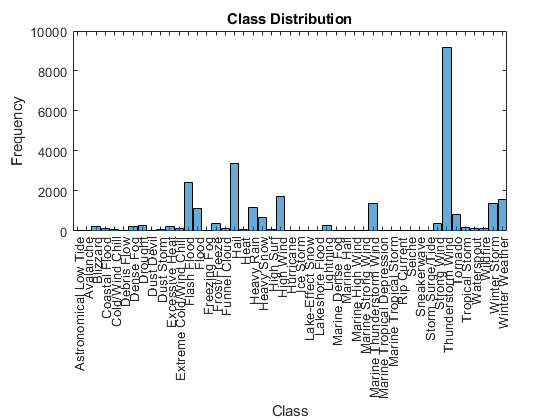
figure

h = histogram(data.event\_type);

xlabel("Class")

ylabel("Frequency")

title("Class Distribution")



数据的类是不平衡的, 几个类包含几个观察点。若要确保可以对数据进行分区, 以便分区包含每个类的观察结果, 请删除出现少于十次的任何类。

从直方图中获取类及其名称的频率计数。

classCounts = h.BinCounts;

classNames = h.Categories;

查找包含少于十个观察点的类, 并从数据中移除这些不常见的类。

idxLowCounts = classCounts < 10;

infrequentClasses = classNames(idxLowCounts);

idxInfrequent = ismember(data.event\_type,infrequentClasses);

data(idxInfrequent,:) = [];

将数据划分为训练分区和已保留的测试集。指定维持百分比为10%。

cvp = cvpartition(data.event\_type,'Holdout',0.1);

dataTrain = data(cvp.training,:);

dataTest = data(cvp.test,:);

从表中提取文本数据和标签。

textDataTrain = dataTrain.event\_narrative;

textDataTest = dataTest.event\_narrative;

YTrain = dataTrain.event\_type;

YTest = dataTest.event\_type;

### 准备用于分析的文本数据

创建先标记和预处理文本数据的函数, 以便用于分析。函数preprocessWeatherNarratives, 按顺序执行以下步骤:

1. 使用lower的文本数据转换为小写.
2. 使用tokenizedDocument标记化文本.
3. 使用erasePunctuation擦除标点.
4. 使用removeStopWords删除停止字词 (如 "and"、"" 和 ") 的列表.
5. 使用removeShortWords删除2个或更少字符的单词.
6. 使用removeLongWords删除具有15个或更多字符的单词.
7. 用normalizeWords Lemmatize 单词.

使用示例预处理函数preprocessWeatherNarratives准备文本数据。

documents = preprocessWeatherNarratives(textDataTrain);

documents(1:5)

ans =

5×1 tokenizedDocument:

(1,1) 5 tokens: large tree down plantersville nettleton

(2,1) 18 tokens: two foot deep standing water develop street winthrop unive…

(3,1) 9 tokens: nws columbia relayed report tree blow down tom hall

(4,1) 10 tokens: medium report two tree blow down i40 old fort area

(5,1) 8 tokens: few tree limb great inches down hwy roseland

从标记化文档创建一个包词模型。

bag = bagOfWords(documents)

bag =

bagOfWords with properties:

Counts: [25316×17458 double]

Vocabulary: [1×17458 string]

NumWords: 17458

NumDocuments: 25316

从单词袋模型中删除不超过两次的单词。删除任何不包含字词的文档, 并删除标签中的相应条目。

bag = removeInfrequentWords(bag,2);

[bag,idx] = removeEmptyDocuments(bag);

YTrain(idx) = [];

bag

bag =

bagOfWords with properties:

Counts: [25315×6515 double]

Vocabulary: [1×6515 string]

NumWords: 6515

NumDocuments: 25315

### 训练监督分类器

使用单词频率计数从词袋模型和标签训练监督分类模型。

使用fitcecoc训练多级线性分类模型。将单词袋模型的Counts属性指定为预测因子, 事件类型标签为响应。指定学习者为线性。这些学习者支持稀疏数据输入。

XTrain = bag.Counts;

mdl = fitcecoc(XTrain,YTrain,'Learners','linear')

mdl =

classreg.learning.classif.CompactClassificationECOC

ResponseName: 'Y'

ClassNames: [1×39 categorical]

ScoreTransform: 'none'

BinaryLearners: {741×1 cell}

CodingMatrix: [39×741 double]

Properties, Methods

为了更好地适应, 您可以尝试指定线性学习者的不同参数。有关线性分类学习者模板的详细信息, 请参阅[templateLinear](https://www.mathworks.com/help/stats/templatelinear.html).

### 测试分类器

使用经过训练的模型预测测试数据的标签, 并计算分类精度。分类精度是模型预测正确的标签的比例。

使用与训练数据相同的预处理步骤预处理测试数据。根据单词包的模型, 将生成的测试文档编码为词频计数矩阵。

documentsTest = preprocessWeatherNarratives(textDataTest);

XTest = encode(bag,documentsTest);

使用经过训练的模型预测测试数据的标签, 并计算分类精度。

YPred = predict(mdl,XTest);

acc = sum(YPred == YTest)/numel(YTest)

acc = 0.8729

### 使用新数据预测

对新天气报告的事件类型进行分类。创建包含新天气报告的字符串数组。

str = [ ...

"A large tree is downed and blocking traffic outside Apple Hill."

"Damage to many car windshields in parking lot."

"Lots of water damage to computer equipment inside the office."];

documentsNew = preprocessWeatherNarratives(str);

XNew = encode(bag,documentsNew);

labelsNew = predict(mdl,XNew)

labelsNew = 3×1 categorical array

Thunderstorm Wind

Thunderstorm Wind

Flash Flood

### 预处理函数示例

函数preprocessWeatherNarratives, 按顺序执行以下步骤:

1. 使用lower的文本数据转换为小写.
2. 使用tokenizedDocument标记化文本.
3. 使用erasePunctuation擦除标点.
4. 使用removeStopWords删除停止字词 (如 "and"、"" 和 ") 的列表.
5. 使用removeShortWords删除2个或更少字符的单词.
6. 使用removeLongWords删除具有15个或更多字符的单词.
7. 用normalizeWords Lemmatize 单词.

function documents = preprocessWeatherNarratives(textData)

% Convert the text data to lowercase.

cleanTextData = lower(textData);

% Tokenize the text.

documents = tokenizedDocument(cleanTextData);

% 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 greater

% characters.

documents = removeShortWords(documents,2);

documents = removeLongWords(documents,15);

% Lemmatize the words.

documents = addPartOfSpeechDetails(documents);

documents = normalizeWords(documents,'Style','lemma');

end

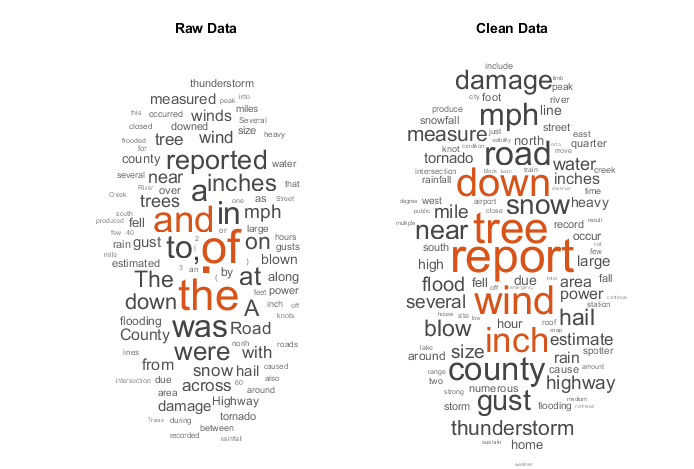
## 准备用于分析的文本数据

此示例演示如何创建清理和预处理文本数据进行分析的函数。

文本数据可能很大, 可能包含大量的噪声, 这会对统计分析产生负面影响。例如, 文本数据可以包含以下内容:

* 情况下的变体, 例如 "新" 和 "新建"
* 单词形式的变体, 例如 "步行" 和 "步行"
* 添加杂色的单词, 例如 "" "和" "的停止词
* 标点符号和特殊字符
* HTML 和 XML 标记

这些字云阐释了应用于天气报告中的一些原始文本数据的词频分析, 以及相同文本数据的预处理版本。



### 加载和提取文本数据

加载示例数据。文件weatherReports.csv包含天气报告, 包括每个事件的文本说明和分类标签。

filename = "weatherReports.csv";

data = readtable(filename,'TextType','string');

从字段event\_narrative中提取文本数据, 并从字段中获取标签数据event\_type.

textData = data.event\_narrative;

labels = data.event\_type;

textData(1:10)

ans = 10×1 string array

"Large tree down between Plantersville and Nettleton."

"One to two feet of deep standing water developed on a street on the Winthrop University campus after more than an inch of rain fell in less than an hour. One vehicle was stalled in the water."

"NWS Columbia relayed a report of trees blown down along Tom Hall St."

"Media reported two trees blown down along I-40 in the Old Fort area."

""

"A few tree limbs greater than 6 inches down on HWY 18 in Roseland."

"Awning blown off a building on Lamar Avenue. Multiple trees down near the intersection of Winchester and Perkins."

"Quarter size hail near Rosemark."

"Tin roof ripped off house on Old Memphis Road near Billings Drive. Several large trees down in the area."

"Powerlines down at Walnut Grove and Cherry Lane roads."

### 创建标记化文档

将文本数据转换为小写。

cleanTextData = lower(textData);

cleanTextData(1:10)

ans = 10×1 string array

"large tree down between plantersville and nettleton."

"one to two feet of deep standing water developed on a street on the winthrop university campus after more than an inch of rain fell in less than an hour. one vehicle was stalled in the water."

"nws columbia relayed a report of trees blown down along tom hall st."

"media reported two trees blown down along i-40 in the old fort area."

""

"a few tree limbs greater than 6 inches down on hwy 18 in roseland."

"awning blown off a building on lamar avenue. multiple trees down near the intersection of winchester and perkins."

"quarter size hail near rosemark."

"tin roof ripped off house on old memphis road near billings drive. several large trees down in the area."

"powerlines down at walnut grove and cherry lane roads."

创建标记化文档的数组。

cleanDocuments = tokenizedDocument(cleanTextData);

cleanDocuments(1:10)

ans =

10×1 tokenizedDocument:

(1,1) 8 tokens: large tree down between plantersville and nettleton .

(2,1) 39 tokens: one to two feet of deep standing water developed on a stre…

(3,1) 14 tokens: nws columbia relayed a report of trees blown down along to…

(4,1) 14 tokens: media reported two trees blown down along i-40 in the old …

(5,1) 0 tokens:

(6,1) 15 tokens: a few tree limbs greater than 6 inches down on hwy 18 in r…

(7,1) 20 tokens: awning blown off a building on lamar avenue . multiple tre…

(8,1) 6 tokens: quarter size hail near rosemark .

(9,1) 21 tokens: tin roof ripped off house on old memphis road near billing…

(10,1) 10 tokens: powerlines down at walnut grove and cherry lane roads .

擦除文档中的标点。

cleanDocuments = erasePunctuation(cleanDocuments);

cleanDocuments(1:10)

ans =

10×1 tokenizedDocument:

(1,1) 7 tokens: large tree down between plantersville and nettleton

(2,1) 37 tokens: one to two feet of deep standing water developed on a stre…

(3,1) 13 tokens: nws columbia relayed a report of trees blown down along to…

(4,1) 13 tokens: media reported two trees blown down along i40 in the old f…

(5,1) 0 tokens:

(6,1) 14 tokens: a few tree limbs greater than 6 inches down on hwy 18 in r…

(7,1) 18 tokens: awning blown off a building on lamar avenue multiple trees…

(8,1) 5 tokens: quarter size hail near rosemark

(9,1) 19 tokens: tin roof ripped off house on old memphis road near billing…

(10,1) 9 tokens: powerlines down at walnut grove and cherry lane roads

"a"、"and"、"to" 和 "(称为停止词)" 之类的词可以为数据添加杂色。使用removeStopWords函数删除停止字词列表。

cleanDocuments = removeStopWords(cleanDocuments);

cleanDocuments(1:10)

ans =

10×1 tokenizedDocument:

(1,1) 5 tokens: large tree down plantersville nettleton

(2,1) 18 tokens: two feet deep standing water developed street winthrop uni…

(3,1) 10 tokens: nws columbia relayed report trees blown down tom hall st

(4,1) 10 tokens: media reported two trees blown down i40 old fort area

(5,1) 0 tokens:

(6,1) 10 tokens: few tree limbs greater 6 inches down hwy 18 roseland

(7,1) 13 tokens: awning blown off building lamar avenue multiple trees down…

(8,1) 5 tokens: quarter size hail near rosemark

(9,1) 16 tokens: tin roof ripped off house old memphis road near billings d…

(10,1) 7 tokens: powerlines down walnut grove cherry lane roads

删除2个或更少字符的单词, 以及具有15个或更大字符的单词。

cleanDocuments = removeShortWords(cleanDocuments,2);

cleanDocuments = removeLongWords(cleanDocuments,15);

cleanDocuments(1:10)

ans =

10×1 tokenizedDocument:

(1,1) 5 tokens: large tree down plantersville nettleton

(2,1) 18 tokens: two feet deep standing water developed street winthrop uni…

(3,1) 9 tokens: nws columbia relayed report trees blown down tom hall

(4,1) 10 tokens: media reported two trees blown down i40 old fort area

(5,1) 0 tokens:

(6,1) 8 tokens: few tree limbs greater inches down hwy roseland

(7,1) 13 tokens: awning blown off building lamar avenue multiple trees down…

(8,1) 5 tokens: quarter size hail near rosemark

(9,1) 16 tokens: tin roof ripped off house old memphis road near billings d…

(10,1) 7 tokens: powerlines down walnut grove cherry lane roads

用normalizeWordsLemmatize 单词。要提高词形还原, 首先使用addPartOfSpeechDetails将部分语音详细信息添加到文档中.

cleanDocuments = addPartOfSpeechDetails(cleanDocuments);

cleanDocuments = normalizeWords(cleanDocuments,'Style','lemma');

cleanDocuments(1:10)

ans =

10×1 tokenizedDocument:

(1,1) 5 tokens: large tree down plantersville nettleton

(2,1) 18 tokens: two foot deep standing water develop street winthrop unive…

(3,1) 9 tokens: nws columbia relayed report tree blow down tom hall

(4,1) 10 tokens: medium report two tree blow down i40 old fort area

(5,1) 0 tokens:

(6,1) 8 tokens: few tree limb great inches down hwy roseland

(7,1) 13 tokens: awning blow off building lamar avenue multiple tree down n…

(8,1) 5 tokens: quarter size hail near rosemark

(9,1) 16 tokens: tin roof rip off house old memphis road near billings driv…

(10,1) 7 tokens: powerlines down walnut grove cherry lane road

### 创建单词袋模型

创建一个词袋模型。

cleanBag = bagOfWords(cleanDocuments)

cleanBag =

bagOfWords with properties:

Counts: [36176×18410 double]

Vocabulary: [1×18410 string]

NumWords: 18410

NumDocuments: 36176

删除在单词袋模型中未出现两次以上的单词。

cleanBag = removeInfrequentWords(cleanBag,2)

cleanBag =

bagOfWords with properties:

Counts: [36176×6952 double]

Vocabulary: [1×6952 string]

NumWords: 6952

NumDocuments: 36176

一些预处理步骤 (如removeInfrequentWords ) 会将空文档留在单词袋模型中。为确保在预处理后没有空文档保留在单词袋模型中, 请使用removeEmptyDocuments作为最后一步。

从单词袋模型中删除空文档和labels中的相应标签.

[cleanBag,idx] = removeEmptyDocuments(cleanBag);

labels(idx) = [];

cleanBag

cleanBag =

bagOfWords with properties:

Counts: [28137×6952 double]

Vocabulary: [1×6952 string]

NumWords: 6952

NumDocuments: 28137

### 创建预处理函数

创建执行预处理的函数可能很有用, 因此您可以以相同的方式准备不同的文本数据集合。例如, 可以使用函数, 以便可以使用与定型数据相同的步骤来预处理新数据。

创建先标记和预处理文本数据的函数, 以便用于分析。函数preprocessWeatherNarratives, 按顺序执行以下步骤:

1. 使用lower的文本数据转换为小写.
2. 使用tokenizedDocument标记化文本.
3. 使用erasePunctuation擦除标点.
4. 使用removeStopWords删除停止字词 (如 "and"、"" 和 ") 的列表.
5. 使用removeShortWords删除2个或更少字符的单词.
6. 使用removeLongWords删除具有15个或更多字符的单词.
7. 用normalizeWords Lemmatize 单词.

使用示例预处理函数preprocessWeatherNarratives准备文本数据。

newText = "A tree is downed outside Apple Hill Drive, Natick";

newDocuments = preprocessWeatherNarratives(newText)

newDocuments =

tokenizedDocument:

7 tokens: tree down outside apple hill drive natick

### 与原始数据进行比较

将预处理后的数据与原始数据进行比较。

rawDocuments = tokenizedDocument(textData);

rawBag = bagOfWords(rawDocuments)

rawBag =

bagOfWords with properties:

Counts: [36176×23302 double]

Vocabulary: [1×23302 string]

NumWords: 23302

NumDocuments: 36176

计算数据的缩减。

numWordsClean = cleanBag.NumWords;

numWordsRaw = rawBag.NumWords;

reduction = 1 - numWordsClean/numWordsRaw

reduction = 0.7017

通过使用 word 云可视化两个单词袋模型, 比较原始数据和已清理数据。

figure

subplot(1,2,1)

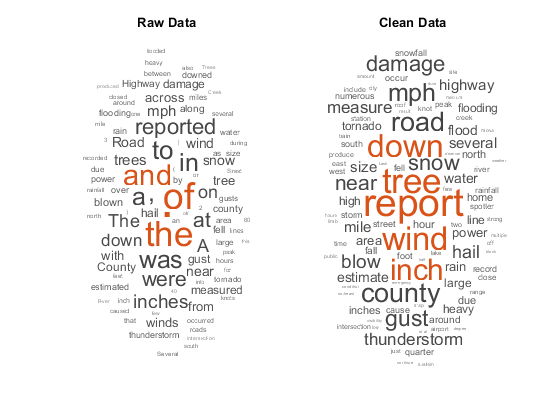
wordcloud(rawBag);

title("Raw Data")

subplot(1,2,2)

wordcloud(cleanBag);

title("Clean Data")



### 预处理函数示例

函数preprocessWeatherNarratives, 按顺序执行以下步骤:

1. 使用lower的文本数据转换为小写.
2. 使用tokenizedDocument标记化文本.
3. 使用erasePunctuation擦除标点.
4. 使用removeStopWords删除停止字词 (如 "and"、"" 和 ") 的列表.
5. 使用removeShortWords删除2个或更少字符的单词.
6. 使用removeLongWords删除具有15个或更多字符的单词.
7. 用normalizeWords Lemmatize 单词.

function [documents] = preprocessWeatherNarratives(textData)

% Convert the text data to lowercase.

cleanTextData = lower(textData);

% Tokenize the text.

documents = tokenizedDocument(cleanTextData);

% 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 greater

% characters.

documents = removeShortWords(documents,2);

documents = removeLongWords(documents,15);

% Lemmatize the words.

documents = addPartOfSpeechDetails(documents);

documents = normalizeWords(documents,'Style','lemma');

end

## 使用主题模型分析文本数据

此示例演示如何使用潜在的 "不分配" (LDA) 主题模型来分析文本数据。

潜在的不分配 (LDA) 模型是一个主题模型, 它发现文档集合中的基础主题, 并在主题中推断单词概率。

若要重现此示例的结果, 请将rng设置为'default'.

rng('default')

### 加载和提取文本数据

加载示例数据。文件weatherReports.csv包含天气报告, 包括每个事件的文本说明和分类标签。

data = readtable("weatherReports.csv",'TextType','string');

head(data)

ans=8×16 table

Time event\_id state event\_type damage\_property damage\_crops begin\_lat begin\_lon end\_lat end\_lon event\_narrative storm\_duration begin\_day end\_day year end\_timestamp

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

22-Jul-2016 16:10:00 6.4433e+05 "MISSISSIPPI" "Thunderstorm Wind" "" "0.00K" 34.14 -88.63 34.122 -88.626 "Large tree down between Plantersville and Nettleton." 00:05:00 22 22 2016 22-Jul-0016 16:15:00

15-Jul-2016 17:15:00 6.5182e+05 "SOUTH CAROLINA" "Heavy Rain" "2.00K" "0.00K" 34.94 -81.03 34.94 -81.03 "One to two feet of deep standing water developed on a street on the Winthrop University campus after more than an inch of rain fell in less than an hour. One vehicle was stalled in the water." 00:00:00 15 15 2016 15-Jul-0016 17:15:00

15-Jul-2016 17:25:00 6.5183e+05 "SOUTH CAROLINA" "Thunderstorm Wind" "0.00K" "0.00K" 35.01 -80.93 35.01 -80.93 "NWS Columbia relayed a report of trees blown down along Tom Hall St." 00:00:00 15 15 2016 15-Jul-0016 17:25:00

16-Jul-2016 12:46:00 6.5183e+05 "NORTH CAROLINA" "Thunderstorm Wind" "0.00K" "0.00K" 35.64 -82.14 35.64 -82.14 "Media reported two trees blown down along I-40 in the Old Fort area." 00:00:00 16 16 2016 16-Jul-0016 12:46:00

15-Jul-2016 14:28:00 6.4332e+05 "MISSOURI" "Hail" "" "" 36.45 -89.97 36.45 -89.97 "" 00:07:00 15 15 2016 15-Jul-0016 14:35:00

15-Jul-2016 16:31:00 6.4332e+05 "ARKANSAS" "Thunderstorm Wind" "" "0.00K" 35.85 -90.1 35.838 -90.087 "A few tree limbs greater than 6 inches down on HWY 18 in Roseland." 00:09:00 15 15 2016 15-Jul-0016 16:40:00

15-Jul-2016 16:03:00 6.4343e+05 "TENNESSEE" "Thunderstorm Wind" "20.00K" "0.00K" 35.056 -89.937 35.05 -89.904 "Awning blown off a building on Lamar Avenue. Multiple trees down near the intersection of Winchester and Perkins." 00:07:00 15 15 2016 15-Jul-0016 16:10:00

15-Jul-2016 17:27:00 6.4344e+05 "TENNESSEE" "Hail" "" "" 35.385 -89.78 35.385 -89.78 "Quarter size hail near Rosemark." 00:05:00 15 15 2016 15-Jul-0016 17:32:00

从字段中提取文本数据event\_narrative.

textData = data.event\_narrative;

textData(1:10)

ans = 10×1 string array

"Large tree down between Plantersville and Nettleton."

"One to two feet of deep standing water developed on a street on the Winthrop University campus after more than an inch of rain fell in less than an hour. One vehicle was stalled in the water."

"NWS Columbia relayed a report of trees blown down along Tom Hall St."

"Media reported two trees blown down along I-40 in the Old Fort area."

""

"A few tree limbs greater than 6 inches down on HWY 18 in Roseland."

"Awning blown off a building on Lamar Avenue. Multiple trees down near the intersection of Winchester and Perkins."

"Quarter size hail near Rosemark."

"Tin roof ripped off house on Old Memphis Road near Billings Drive. Several large trees down in the area."

"Powerlines down at Walnut Grove and Cherry Lane roads."

### 准备用于分析的文本数据

创建先标记和预处理文本数据的函数, 以便用于分析。函数preprocessWeatherNarratives, 按顺序执行以下步骤:

1. 使用lower的文本数据转换为小写.
2. 使用tokenizedDocument标记化文本.
3. 使用erasePunctuation擦除标点.
4. 使用removeStopWords删除停止字词 (如 "and"、"" 和 ") 的列表.
5. 使用removeShortWords删除2个或更少字符的单词.
6. 使用removeLongWords删除具有15个或更多字符的单词.
7. 用normalizeWords Lemmatize 单词.

使用示例预处理函数preprocessWeatherNarratives准备文本数据。

documents = preprocessWeatherNarratives(textData);

documents(1:5)

ans =

5×1 tokenizedDocument:

(1,1) 5 tokens: large tree down plantersville nettleton

(2,1) 18 tokens: two foot deep standing water develop street winthrop unive…

(3,1) 9 tokens: nws columbia relayed report tree blow down tom hall

(4,1) 10 tokens: medium report two tree blow down i40 old fort area

(5,1) 0 tokens:

从标记化文档创建一个包词模型。

bag = bagOfWords(documents)

bag =

bagOfWords with properties:

Counts: [36176×18410 double]

Vocabulary: [1×18410 string]

NumWords: 18410

NumDocuments: 36176

从单词袋模型中删除单词, 这些词的总数不超过两次。删除包含单词袋模型中没有单词的任何文档。

bag = removeInfrequentWords(bag,2);

bag = removeEmptyDocuments(bag)

bag =

bagOfWords with properties:

Counts: [28137×6952 double]

Vocabulary: [1×6952 string]

NumWords: 6952

NumDocuments: 28137

### 适合 LDA 模型

适合7个主题的 LDA 模型。有关演示如何选择主题数量的示例, 请参阅[选择 LDA 模型的主题数](https://www.mathworks.com/help/textanalytics/ug/choose-number-of-topics-for-LDA-model.html)。要抑制详细输出, 请将'Verbose'设置为0。

numTopics = 7;

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

如果您有一个大型数据集, 那么随机近似变分贝叶斯求解器通常更适合, 因为它可以在较少的数据传递中容纳好模型。fitlda的默认解算器 (折叠吉布斯采样) 可以更准确地花费更长的运行时间。要使用随机近似变分贝叶斯, 请将'Solver'选项设置为'savb'。有关演示如何比较 LDA 求解器的示例, 请参阅[使用主题模型分析文本数据](https://www.mathworks.com/help/textanalytics/ug/analyze-text-data-using-topic-models.html).

### 使用 Word 云可视化主题

您可以使用 word 云来查看每个主题中具有最高概率的单词。使用 word 云可视化前四个主题。

figure;

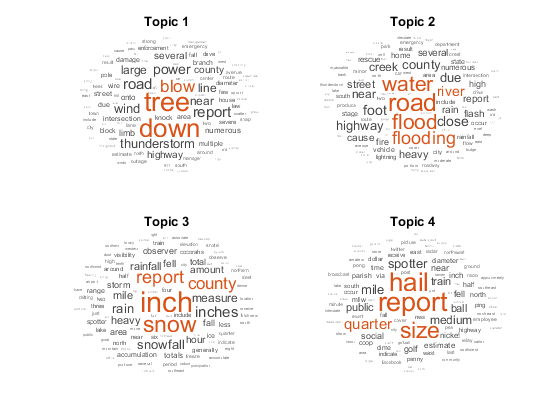
for topicIdx = 1:4

subplot(2,2,topicIdx)

wordcloud(mdl,topicIdx);

title("Topic " + topicIdx)

end



### 查看文档中主题的混合

使用transform将文档转换为主题概率的向量。

newDocument = tokenizedDocument("A tree is downed outside Apple Hill Drive, Natick");

topicMixture = transform(mdl,newDocument);

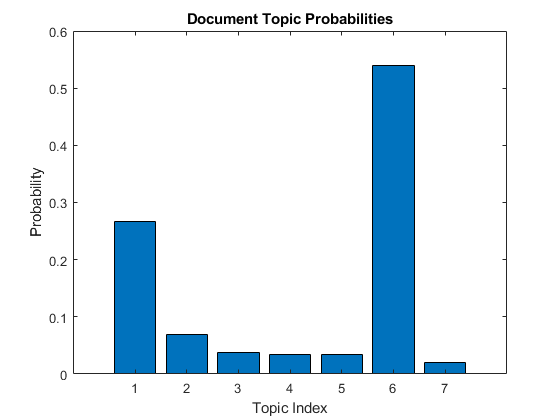
figure

bar(topicMixture)

xlabel("Topic Index")

ylabel("Probability")

title("Document Topic Probabilities")



使用堆积条形图可视化多个主题混合物。可视化前5个输入文档的主题混合。

figure

topicMixtures = transform(mdl,documents(1:5));

barh(topicMixtures(1:5,:),'stacked')

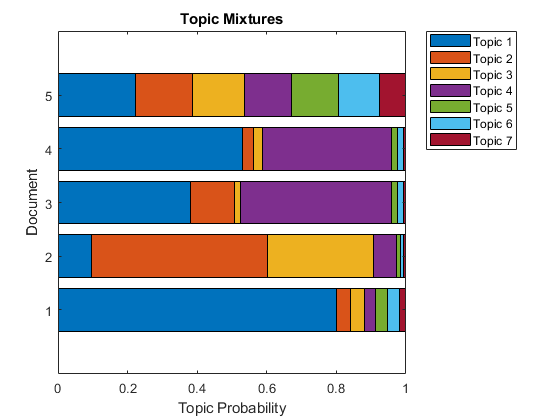
xlim([0 1])

title("Topic Mixtures")

xlabel("Topic Probability")

ylabel("Document")

legend("Topic " + string(1:numTopics),'Location','northeastoutside')



### 预处理函数示例

函数preprocessWeatherNarratives, 按顺序执行以下步骤:

1. 使用lower的文本数据转换为小写.
2. 使用tokenizedDocument标记化文本.
3. 使用erasePunctuation擦除标点.
4. 使用removeStopWords删除停止字词 (如 "and"、"" 和 ") 的列表.
5. 使用removeShortWords删除2个或更少字符的单词.
6. 使用removeLongWords删除具有15个或更多字符的单词.
7. 用normalizeWords Lemmatize 单词.

function [documents] = preprocessWeatherNarratives(textData)

% Convert the text data to lowercase.

cleanTextData = lower(textData);

% Tokenize the text.

documents = tokenizedDocument(cleanTextData);

% 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 greater

% characters.

documents = removeShortWords(documents,2);

documents = removeLongWords(documents,15);

% Lemmatize the words.

documents = addPartOfSpeechDetails(documents);

documents = normalizeWords(documents,'Style','lemma');

end

# 文本数据准备

## 从文件中提取文本数据

此示例演示如何从文本、HTML、微软 Word、PDF、CSV 和 Microsoft Excel®文件中提取文本数据, 并将其导入 MATLAB®进行分析。

通常, 将文本数据导入 MATLAB 最简单的方法是使用extractFileText函数。此函数从文本、PDF、HTML 和 Microsoft Word 文件中提取文本数据。要从 CSV 和 excel 文件导入文本, 请使用readtable。要从 HTML 代码中提取文本, 请使用extractHTMLText。要从 PDF 表单读取数据, 请使用readPDFFormData.

### 文本文件

使用extractFileText从sonnets.txt中提取文本。该文件sonnets.txt包含莎士比亚的诗的纯文本。

filename = "sonnets.txt";

str = extractFileText(filename);

通过提取两个标题 "I" 和 "II " 之间的文本来查看第一个诗歌".

start = " I" + newline;

fin = " II";

sonnet1 = extractBetween(str,start,fin)

sonnet1 =

"

From fairest creatures we desire increase,

That thereby beauty's rose might never die,

But as the riper should by time decease,

His tender heir might bear his memory:

But thou, contracted to thine own bright eyes,

Feed'st thy light's flame with self-substantial fuel,

Making a famine where abundance lies,

Thy self thy foe, to thy sweet self too cruel:

Thou that art now the world's fresh ornament,

And only herald to the gaudy spring,

Within thine own bud buriest thy content,

And tender churl mak'st waste in niggarding:

Pity the world, or else this glutton be,

To eat the world's due, by the grave and thee.

"

### 微软 Word 文档

使用extractFileText提取文字sonnets.docx文件exampleSonnets.docx在微软 Word 文档中包含莎士比亚的诗歌。

filename = "exampleSonnets.docx";

str = extractFileText(filename);

通过提取两个标题 "II" 和 "III " 之间的文本来查看第二个诗歌".

start = " II" + newline;

fin = " III";

sonnet2 = extractBetween(str,start,fin)

sonnet2 =

"

When forty winters shall besiege thy brow,

And dig deep trenches in thy beauty's field,

Thy youth's proud livery so gazed on now,

Will be a tatter'd weed of small worth held:

Then being asked, where all thy beauty lies,

Where all the treasure of thy lusty days;

To say, within thine own deep sunken eyes,

Were an all-eating shame, and thriftless praise.

How much more praise deserv'd thy beauty's use,

If thou couldst answer 'This fair child of mine

Shall sum my count, and make my old excuse,'

Proving his beauty by succession thine!

This were to be new made when thou art old,

And see thy blood warm when thou feel'st it cold.

"

示例 Microsoft Word 文档在每行之间使用两个换行符。若要用单个换行符替换这些字符, 请使用 "replace" 函数。

sonnet2 = replace(sonnet2,[newline newline],newline)

sonnet2 =

"

When forty winters shall besiege thy brow,

And dig deep trenches in thy beauty's field,

Thy youth's proud livery so gazed on now,

Will be a tatter'd weed of small worth held:

Then being asked, where all thy beauty lies,

Where all the treasure of thy lusty days;

To say, within thine own deep sunken eyes,

Were an all-eating shame, and thriftless praise.

How much more praise deserv'd thy beauty's use,

If thou couldst answer 'This fair child of mine

Shall sum my count, and make my old excuse,'

Proving his beauty by succession thine!

This were to be new made when thou art old,

And see thy blood warm when thou feel'st it cold.

"

### PDF 文件

从 pdf 格式的 pdf 文档和数据中提取文本。

使用extractFileText从诗中提取文本sonnets.pdf文件exampleSonnets.pdf在 pdf 中包含莎士比亚的诗歌。

filename = "exampleSonnets.pdf";

str = extractFileText(filename);

通过提取两个标题 "III" 和 "IV" 之间的文本来查看第三个诗歌。此 PDF 在每个换行符之前都有空格。

start = " III " + newline;

fin = "IV";

sonnet3 = extractBetween(str,start,fin)

sonnet3 =

"

Look in thy glass and tell the face thou viewest

Now is the time that face should form another;

Whose fresh repair if now thou not renewest,

Thou dost beguile the world, unbless some mother.

For where is she so fair whose unear'd womb

Disdains the tillage of thy husbandry?

Or who is he so fond will be the tomb,

Of his self-love to stop posterity?

Thou art thy mother's glass and she in thee

Calls back the lovely April of her prime;

So thou through windows of thine age shalt see,

Despite of wrinkles this thy golden time.

But if thou live, remember'd not to be,

Die single and thine image dies with thee.

"

要从 PDF 表单读取文本数据, 请使用readPDFFormData。该函数返回包含 PDF 表单域中数据的结构。

filename = "weatherReportForm1.pdf";

data = readPDFFormData(filename)

data = struct with fields:

event\_type: "Thunderstorm Wind"

event\_narrative: "Large tree down between Plantersville and Nettleton."

### Html

从 html 文件、html 代码和 web 中提取文本。

要从保存的 HTML 文件中提取文本数据, 请使用extractFileText.

filename = "exampleSonnets.html";

str = extractFileText(filename);

通过提取两个标题"IV"和"V"之间的文本来查看第四诗歌.

start = newline + "IV" + newline;

fin = newline + "V" + newline;

sonnet4 = extractBetween(str,start,fin)

sonnet4 =

"

Unthrifty loveliness, why dost thou spend

Upon thy self thy beauty's legacy?

Nature's bequest gives nothing, but doth lend,

And being frank she lends to those are free:

Then, beauteous niggard, why dost thou abuse

The bounteous largess given thee to give?

Profitless usurer, why dost thou use

So great a sum of sums, yet canst not live?

For having traffic with thy self alone,

Thou of thy self thy sweet self dost deceive:

Then how when nature calls thee to be gone,

What acceptable audit canst thou leave?

Thy unused beauty must be tombed with thee,

Which, used, lives th' executor to be.

"

若要从包含 HTML 代码的字符串中提取文本数据, 请使用extractHTMLText.

code = "<html><body><h1>THE SONNETS</h1><p>by William Shakespeare</p></body></html>";

str = extractHTMLText(code)

str =

"THE SONNETS

by William Shakespeare"

若要从网页中提取文本数据, 请首先使用webread读取 HTML 代码, 然后使用extractHTMLText.

url = "https://www.mathworks.com/help/textanalytics";

code = webread(url);

str = extractHTMLText(code)

str =

'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.'

若要查找 HTML 代码的特定元素, 请使用htmlTree分析代码并使用findElement。分析 HTML 代码并查找所有超链接。超链接是具有元素名称"A"的节点.

tree = htmlTree(code);

selector = "A";

subtrees = findElement(tree,selector);

查看前10个子树并使用extractHTMLText提取文本.

subtrees(1:10)

ans =

10×1 htmlTree:

(1,1) <A class="svg\_link navbar-brand" href="https://www.mathworks.com?s\_ti…

(2,1) <A class="mwa-nav\_login" href="https://www.mathworks.com/login?uri=ht…

(3,1) <A href="https://www.mathworks.com/products.html?s\_tid=gn\_ps">Product…

(4,1) <A href="https://www.mathworks.com/solutions.html?s\_tid=gn\_sol">Solut…

(5,1) <A href="https://www.mathworks.com/academia.html?s\_tid=gn\_acad">Acade…

(6,1) <A href="https://www.mathworks.com/support.html?s\_tid=gn\_supp">Suppor…

(7,1) <A href="https://www.mathworks.com/matlabcentral/?s\_tid=gn\_mlc">Commu…

(8,1) <A href="https://www.mathworks.com/company/events.html?s\_tid=gn\_ev">E…

(9,1) <A href="https://www.mathworks.com/company/aboutus/contact\_us.html?s\_…

(10,1) <A href="https://www.mathworks.com/store?s\_cid=store\_top\_nav&amp;s\_ti…

str = extractHTMLText(subtrees);

查看前10个超链接的提取文本。

str(1:10)

ans = 10×1 string array

""

"Sign In"

"Products"

"Solutions"

"Academia"

"Support"

"Community"

"Events"

"Contact Us"

"How to Buy"

若要获取链接目标, 请使用getAttributes并指定属性"href" (超链接引用)。获取前10个子树的链接目标。

attr = "href";

str = getAttribute(subtrees(1:10),attr)

str = 10×1 string array

"https://www.mathworks.com?s\_tid=gn\_logo"

"https://www.mathworks.com/login?uri=http://www.mathworks.com/help/textanalytics/index.html"

"https://www.mathworks.com/products.html?s\_tid=gn\_ps"

"https://www.mathworks.com/solutions.html?s\_tid=gn\_sol"

"https://www.mathworks.com/academia.html?s\_tid=gn\_acad"

"https://www.mathworks.com/support.html?s\_tid=gn\_supp"

"https://www.mathworks.com/matlabcentral/?s\_tid=gn\_mlc"

"https://www.mathworks.com/company/events.html?s\_tid=gn\_ev"

"https://www.mathworks.com/company/aboutus/contact\_us.html?s\_tid=gn\_cntus"

"https://www.mathworks.com/store?s\_cid=store\_top\_nav&s\_tid=gn\_store"

### CSV 和微软 Excel 文件

若要从 CSV 和 Microsoft Excel 文件中提取文本数据, 请使用readtable并从它返回的表中提取文本数据。

从 weatherReports 的events\_narrative列中提取文本weatherReports.csv.

T = readtable('weatherReports.csv','TextType','string');

head(T)

ans=8×16 table

Time event\_id state event\_type damage\_property damage\_crops begin\_lat begin\_lon end\_lat end\_lon event\_narrative storm\_duration begin\_day end\_day year end\_timestamp

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

22-Jul-2016 16:10:00 6.4433e+05 "MISSISSIPPI" "Thunderstorm Wind" "" "0.00K" 34.14 -88.63 34.122 -88.626 "Large tree down between Plantersville and Nettleton." 00:05:00 22 22 2016 22-Jul-0016 16:15:00

15-Jul-2016 17:15:00 6.5182e+05 "SOUTH CAROLINA" "Heavy Rain" "2.00K" "0.00K" 34.94 -81.03 34.94 -81.03 "One to two feet of deep standing water developed on a street on the Winthrop University campus after more than an inch of rain fell in less than an hour. One vehicle was stalled in the water." 00:00:00 15 15 2016 15-Jul-0016 17:15:00

15-Jul-2016 17:25:00 6.5183e+05 "SOUTH CAROLINA" "Thunderstorm Wind" "0.00K" "0.00K" 35.01 -80.93 35.01 -80.93 "NWS Columbia relayed a report of trees blown down along Tom Hall St." 00:00:00 15 15 2016 15-Jul-0016 17:25:00

16-Jul-2016 12:46:00 6.5183e+05 "NORTH CAROLINA" "Thunderstorm Wind" "0.00K" "0.00K" 35.64 -82.14 35.64 -82.14 "Media reported two trees blown down along I-40 in the Old Fort area." 00:00:00 16 16 2016 16-Jul-0016 12:46:00

15-Jul-2016 14:28:00 6.4332e+05 "MISSOURI" "Hail" "" "" 36.45 -89.97 36.45 -89.97 "" 00:07:00 15 15 2016 15-Jul-0016 14:35:00

15-Jul-2016 16:31:00 6.4332e+05 "ARKANSAS" "Thunderstorm Wind" "" "0.00K" 35.85 -90.1 35.838 -90.087 "A few tree limbs greater than 6 inches down on HWY 18 in Roseland." 00:09:00 15 15 2016 15-Jul-0016 16:40:00

15-Jul-2016 16:03:00 6.4343e+05 "TENNESSEE" "Thunderstorm Wind" "20.00K" "0.00K" 35.056 -89.937 35.05 -89.904 "Awning blown off a building on Lamar Avenue. Multiple trees down near the intersection of Winchester and Perkins." 00:07:00 15 15 2016 15-Jul-0016 16:10:00

15-Jul-2016 17:27:00 6.4344e+05 "TENNESSEE" "Hail" "" "" 35.385 -89.78 35.385 -89.78 "Quarter size hail near Rosemark." 00:05:00 15 15 2016 15-Jul-0016 17:32:00

str = T.event\_narrative;

str(1:10)

ans = 10×1 string array

"Large tree down between Plantersville and Nettleton."

"One to two feet of deep standing water developed on a street on the Winthrop University campus after more than an inch of rain fell in less than an hour. One vehicle was stalled in the water."

"NWS Columbia relayed a report of trees blown down along Tom Hall St."

"Media reported two trees blown down along I-40 in the Old Fort area."

""

"A few tree limbs greater than 6 inches down on HWY 18 in Roseland."

"Awning blown off a building on Lamar Avenue. Multiple trees down near the intersection of Winchester and Perkins."

"Quarter size hail near Rosemark."

"Tin roof ripped off house on Old Memphis Road near Billings Drive. Several large trees down in the area."

"Powerlines down at Walnut Grove and Cherry Lane roads."

### 从多个文件中提取文本

如果文本数据包含在文件夹中的多个文件中, 则可以使用文件数据存储将文本数据导入到 MATLAB 中。

为示例诗歌文本文件创建文件数据存储。示例文件被命名为 "exampleSonnetN.txt", 其中N是诗歌的编号。使用通配符 "\*" 指定文件名以查找此结构的所有文件名。若要指定要extractFileText的读取函数, 请使用函数句柄将此函数输入fileDatastore 。

fds = fileDatastore('exampleSonnet\*.txt','ReadFcn',@extractFileText)

fds =

FileDatastore with properties:

Files: {

' ...\Documents\MATLAB\examples\textanalytics-ex15735454\exampleSonnet1.txt';

' ...\Documents\MATLAB\examples\textanalytics-ex15735454\exampleSonnet2.txt';

' ...\Documents\MATLAB\examples\textanalytics-ex15735454\exampleSonnet3.txt'

... and 1 more

}

UniformRead: 0

ReadFcn: @extractFileText

AlternateFileSystemRoots: {}

循环查看数据存储中的文件并读取每个文本文件。

str = [];

while hasdata(fds)

textData = read(fds);

str = [str; textData];

end

查看提取的文本。

str

str = 4×1 string array

" From fairest creatures we desire increase,↵ That thereby beauty's rose might never die,↵ But as the riper should by time decease,↵ His tender heir might bear his memory:↵ But thou, contracted to thine own bright eyes,↵ Feed'st thy light's flame with self-substantial fuel,↵ Making a famine where abundance lies,↵ Thy self thy foe, to thy sweet self too cruel:↵ Thou that art now the world's fresh ornament,↵ And only herald to the gaudy spring,↵ Within thine own bud buriest thy content,↵ And tender churl mak'st waste in niggarding:↵ Pity the world, or else this glutton be,↵ To eat the world's due, by the grave and thee."

" When forty winters shall besiege thy brow,↵ And dig deep trenches in thy beauty's field,↵ Thy youth's proud livery so gazed on now,↵ Will be a tatter'd weed of small worth held:↵ Then being asked, where all thy beauty lies,↵ Where all the treasure of thy lusty days;↵ To say, within thine own deep sunken eyes,↵ Were an all-eating shame, and thriftless praise.↵ How much more praise deserv'd thy beauty's use,↵ If thou couldst answer 'This fair child of mine↵ Shall sum my count, and make my old excuse,'↵ Proving his beauty by succession thine!↵ This were to be new made when thou art old,↵ And see thy blood warm when thou feel'st it cold."

" Look in thy glass and tell the face thou viewest↵ Now is the time that face should form another;↵ Whose fresh repair if now thou not renewest,↵ Thou dost beguile the world, unbless some mother.↵ For where is she so fair whose unear'd womb↵ Disdains the tillage of thy husbandry?↵ Or who is he so fond will be the tomb,↵ Of his self-love to stop posterity?↵ Thou art thy mother's glass and she in thee↵ Calls back the lovely April of her prime;↵ So thou through windows of thine age shalt see,↵ Despite of wrinkles this thy golden time.↵ But if thou live, remember'd not to be,↵ Die single and thine image dies with thee."

" Unthrifty loveliness, why dost thou spend↵ Upon thy self thy beauty's legacy?↵ Nature's bequest gives nothing, but doth lend,↵ And being frank she lends to those are free:↵ Then, beauteous niggard, why dost thou abuse↵ The bounteous largess given thee to give?↵ Profitless usurer, why dost thou use↵ So great a sum of sums, yet canst not live?↵ For having traffic with thy self alone,↵ Thou of thy self thy sweet self dost deceive:↵ Then how when nature calls thee to be gone,↵ What acceptable audit canst thou leave?↵ Thy unused beauty must be tombed with thee,↵ Which, used, lives th' executor to be."

## 分析包含表情符号的文本数据

此示例演示如何分析包含表情符号的文本数据。

表情符号是在文本中以内联形式出现的图形标志。在移动设备 (如智能手机和平板电脑) 上书写文本时, 人们使用表情符号来保持文本的简短并传达情感和情感。

您还可以使用表情符号来分析文本数据。例如, 使用它们来标识文本的相关字符串或可视化文本的情感或情感。

使用文本数据时, 表情符号可能会表现得不可预知。根据您的系统字体, 系统可能无法正确显示某些表情符号。因此, 如果表情符号显示不正确, 则数据不一定会丢失。您的系统可能无法显示当前字体中的表情符号。

### 作曲表情符号

在大多数情况下, 您可以从文件中读取表情符号 (例如, 通过使用extractFileText、 extractHTMLText或readtable), 或者直接将其复制并粘贴到 MATLAB®中。否则, 您必须使用 Unicode UTF16 代码单元撰写表情符号。

一些表情符号由多个 Unicode UTF16 代码单元组成。例如, "带太阳镜的笑脸" 表情符号 (带有代码点 U+1F60E 的😎) 是一个单一字形, 但包含两个 UTF16 代码单元"D83D"和"DE0E"。使用compose函数创建包含此表情符号的字符串, 并指定带有前缀"\x"的两个代码单元.

emoji = compose("\xD83D\xDE0E")

emoji =

"😎"

首先获取一个表情符号的 Unicode UTF16 代码单位。使用char获取表情符号的数字表示形式, 然后使用dec2hex获取相应的十六进制值。

codeUnits = dec2hex(char(emoji))

codeUnits = 2x4 char array

'D83D'

'DE0E'

使用带有空分隔符的strjoin函数重建组合字符串"".

formatSpec = strjoin("\x" + codeUnits,"")

formatSpec =

"\xD83D\xDE0E"

emoji = compose(formatSpec)

emoji =

"😎"

### 导入文本数据

使用readtable提取文件weekendUpdates.xlsx中的文本数据。文件weekendUpdates.xlsx包含包含标签"#weekend"和"#vacation"的状态更新.

filename = "weekendUpdates.xlsx";

tbl = readtable(filename,'TextType','string');

head(tbl)

ans=8×2 table

ID TextData

\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1 "Happy anniversary! ❤ Next stop: Paris! ✈ #vacation"

2 "Haha, BBQ on the beach, engage smug mode! 😍 😎 ❤ 🎉 #vacation"

3 "getting ready for Saturday night 🍕 #yum #weekend 😎"

4 "Say it with me - I NEED A #VACATION!!! ☹"

5 "😎 Chilling 😎 at home for the first time in ages…This is the life! 👍 #weekend"

6 "My last #weekend before the exam 😢 👎."

7 "can’t believe my #vacation is over 😢 so unfair"

8 "Can’t wait for tennis this #weekend 🎾🍓🥂 😀"

从字段TextData中提取文本数据并查看前几个状态更新。

textData = tbl.TextData;

textData(1:5)

ans = 5x1 string array

"Happy anniversary! ❤ Next stop: Paris! ✈ #vacation"

"Haha, BBQ on the beach, engage smug mode! 😍 😎 ❤ 🎉 #vacation"

"getting ready for Saturday night 🍕 #yum #weekend 😎"

"Say it with me - I NEED A #VACATION!!! ☹"

"😎 Chilling 😎 at home for the first time in ages…This is the life! 👍 #weekend"

在 word 云中可视化文本数据。

figure

wordcloud(textData);



### 按表情符号筛选文本数据

使用contains函数标识包含特定表情符号的状态更新。查找包含 "太阳镜笑脸" 表情符号 (😎代码 U+1F60E) 的文档的索引。此表情符号包含两个 Unicode UTF16 代码单元"D83D"和 "DE0E".

emoji = compose("\xD83D\xDE0E");

idx = contains(textData,emoji);

textDataSunglasses = textData(idx);

textDataSunglasses(1:5)

ans = 5x1 string array

"Haha, BBQ on the beach, engage smug mode! 😍 😎 ❤ 🎉 #vacation"

"getting ready for Saturday night 🍕 #yum #weekend 😎"

"😎 Chilling 😎 at home for the first time in ages…This is the life! 👍 #weekend"

"🎉 Check the out-of-office crew, we are officially ON #VACATION!! 😎"

"Who needs a #vacation when the weather is this good ☀ 😎"

在 word 云中可视化提取的文本数据。

figure

wordcloud(textDataSunglasses);



### 提取和可视化表情符号

使用单词云可视化文本数据中的所有表情符号。

提取表情符号。首先使用tokenizedDocument标记化文本, 然后查看前几个文档。

documents = tokenizedDocument(textData);

documents(1:5)

ans =

5x1 tokenizedDocument:

11 tokens: Happy anniversary ! ❤ Next stop : Paris ! ✈ #vacation

16 tokens: Haha , BBQ on the beach , engage smug mode ! 😍 😎 ❤ 🎉 #vacation

9 tokens: getting ready for Saturday night 🍕 #yum #weekend 😎

13 tokens: Say it with me - I NEED A #VACATION ! ! ! ☹

19 tokens: 😎 Chilling 😎 at home for the first time in ages … This is the life ! 👍 #weekend

tokenizedDocument函数自动检测表情符号并分配令牌类型"emoji"。使用tokenDetails函数查看文档的前几个令牌详细信息。

tdetails = tokenDetails(documents);

head(tdetails)

ans=8×5 table

Token DocumentNumber LineNumber Type Language

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

"Happy" 1 1 letters en

"anniversary" 1 1 letters en

"!" 1 1 punctuation en

"❤" 1 1 emoji en

"Next" 1 1 letters en

"stop" 1 1 letters en

":" 1 1 punctuation en

"Paris" 1 1 letters en

通过提取带有令牌类型"emoji"的标记并将其输入到wordcloud函数中, 可视化词云中的表情符号。

idx = tdetails.Type == "emoji";

tokens = tdetails.Token(idx);

figure

wordcloud(tokens);

title("Emojis")



## 分析 HTML 和提取文本内容

此示例演示如何分析 HTML 代码并从特定元素中提取文本内容。

### 解析 HTML 代码

使用webread从 URL https://www.mathworks.com/help/textanalytics读取 HTML 代码.

url = "https://www.mathworks.com/help/textanalytics";

code = webread(url);

使用htmlTree分析 HTML 代码.

tree = htmlTree(code);

查看树的 HTML 元素名称。

tree.Name

ans =

"HTML"

查看树的子元素。孩子是tree的子树.

tree.Children

ans =

4×1 htmlTree:

" "

<HEAD><TITLE>Text Analytics Toolbox Documentation</TITLE><META charset="utf-8"/><META content="width=device-width, initial-scale=1.0" name="viewport"/><META content="IE=edge" http-equiv="X-UA-Compatible"/><LINK href="/includes\_content/responsive/css/bootstrap/bootstrap.min.css" rel="stylesheet" type="text/css"/><LINK href="/includes\_content/responsive/css/site6.css?20180314" rel="stylesheet" type="text/css"/><LINK href="/includes\_content/responsive/css/site6\_lg.css?20180314" media="screen and (min-width: 1200px)" rel="stylesheet"/><LINK href="/includes\_content/responsive/css/site6\_md.css?20180314" media="screen and (min-width: 992px) and (max-width: 1199px)" rel="stylesheet"/><LINK href="/includes\_content/responsive/css/site6\_sm+xs.css?20180314" media="screen and (max-width: 991px)" rel="stylesheet"/><LINK href="/includes\_content/responsive/css/site6\_sm.css?20180314" media="screen and (min-width: 768px) and (max-width: 991px)" rel="stylesheet"/><LINK href="/includes\_content/responsive/…

" "

<BODY id="responsive\_offcanvas"><!-- Mobile TopNav: Start --><DIV class="header visible-xs visible-sm" id="header\_mobile" translate="no"><NAV class="navbar navbar-default" role="navigation"><DIV class="container-fluid"><DIV class="row"><DIV class="col-xs-12"><DIV class="navbar-header"><BUTTON class="navbar-toggle topnav\_toggle" data-target="#topnav\_collapse" data-toggle="collapse" type="button"><SPAN class="sr-only">Toggle Main Navigation</SPAN><SPAN class="icon-menu"/></BUTTON><A class="svg\_link navbar-brand" href="https://www.mathworks.com?s\_tid=gn\_logo"><IMG alt="MathWorks" class="mw\_logo" src="/images/responsive/global/pic-header-mathworks-logo.svg"/></A></DIV></DIV></DIV><DIV class="row visible-xs visible-sm"><DIV class="col-xs-12"><DIV class="navbar-collapse collapse" id="topnav\_collapse"><UL class="nav navbar-nav" id="topnav"><LI class="headernav\_login"><A class="mwa-nav\_login" href="https://www.mathworks.com/login?uri=http://www.mathworks.com/help/textanalytics/index.html">Sign…

### 从 HTML 树中提取文本

要直接从 HTML 树中提取文本, 请使用extractHTMLText.

str = extractHTMLText(tree)

str =

"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."

### 查找 HTML 元素

若要查找 HTML 树的特定元素, 请使用findElement。查找 HTML 树中的所有超链接。在 HTML 中, 超链接使用 "A" 标记。

selector = "A";

subtrees = findElement(tree,selector);

查看前几个子树。

subtrees(1:20)

ans =

20×1 htmlTree:

<A class="svg\_link navbar-brand" href="https://www.mathworks.com?s\_tid=gn\_logo"><IMG alt="MathWorks" class="mw\_logo" src="/images/responsive/global/pic-header-mathworks-logo.svg"/></A>

<A class="mwa-nav\_login" href="https://www.mathworks.com/login?uri=http://www.mathworks.com/help/textanalytics/index.html">Sign In</A>

<A href="https://www.mathworks.com/products.html?s\_tid=gn\_ps">Products</A>

<A href="https://www.mathworks.com/solutions.html?s\_tid=gn\_sol">Solutions</A>

<A href="https://www.mathworks.com/academia.html?s\_tid=gn\_acad">Academia</A>

<A href="https://www.mathworks.com/support.html?s\_tid=gn\_supp">Support</A>

<A href="https://www.mathworks.com/matlabcentral/?s\_tid=gn\_mlc">Community</A>

<A href="https://www.mathworks.com/company/events.html?s\_tid=gn\_ev">Events</A>

<A href="https://www.mathworks.com/company/aboutus/contact\_us.html?s\_tid=gn\_cntus">Contact Us</A>

<A href="https://www.mathworks.com/store?s\_cid=store\_top\_nav&amp;s\_tid=gn\_store">How to Buy</A>

<A href="https://www.mathworks.com/company/aboutus/contact\_us.html?s\_tid=gn\_cntus">Contact Us</A>

<A href="https://www.mathworks.com/store?s\_cid=store\_top\_nav&amp;s\_tid=gn\_store">How to Buy</A>

<A class="mwa-nav\_login" href="https://www.mathworks.com/login?uri=http://www.mathworks.com/help/textanalytics/index.html">Sign In</A>

<A class="svg\_link pull-left" href="https://www.mathworks.com?s\_tid=gn\_logo"><IMG alt="MathWorks" class="mw\_logo" src="/images/responsive/global/pic-header-mathworks-logo.svg"/></A>

<A href="https://www.mathworks.com/products.html?s\_tid=gn\_ps">Products</A>

<A href="https://www.mathworks.com/solutions.html?s\_tid=gn\_sol">Solutions</A>

<A href="https://www.mathworks.com/academia.html?s\_tid=gn\_acad">Academia</A>

<A href="https://www.mathworks.com/support.html?s\_tid=gn\_supp">Support</A>

<A href="https://www.mathworks.com/matlabcentral/?s\_tid=gn\_mlc">Community</A>

<A href="https://www.mathworks.com/company/events.html?s\_tid=gn\_ev">Events</A>

从超链接的文本创建一个词云。

str = extractHTMLText(subtrees);

figure

wordcloud(str);

title("Hyperlinks")



### 获取 HTML 属性

从 HTML 树中的段落元素获取类属性。

subtrees = findElement(tree,'p');

attr = "class";

str = getAttribute(subtrees,attr)

str = 21×1 string array

<missing>

<missing>

"add\_margin\_5"

<missing>

<missing>

<missing>

<missing>

<missing>

"category\_desc"

"category\_desc"

"category\_desc"

"category\_desc"

<missing>

<missing>

<missing>

"text-center"

<missing>

<missing>

<missing>

"copyright"

<missing>

使用类"category\_desc"从段落元素中包含的文本创建单词云.

subtrees = findElement(tree,'p.category\_desc');

str = extractHTMLText(subtrees);

figure

wordcloud(str);



## 分析日语文本数据

此示例演示如何使用主题模型导入、准备和分析日语文本数据。

日语文本数据可能很大, 可能包含大量的噪声, 从而对统计分析产生负面影响。例如, 文本数据可以包含以下内容:

* 词形式的变体。例如, "難しい" ("很困难") 和 "難しかった" ("很困难")
* 添加杂色的单词。例如, 停止单词, 如 "あそこ" ("在那里")、"あたり" ("周围") 和 "あちら" ("有")
* 标点符号和特殊字符

这些字云阐释了从 "吾輩は猫である" 到夏目漱石的一些原始文本数据以及同一文本数据的预处理版本所应用的词频分析。



本示例首先演示如何导入和准备日文文本数据, 然后演示如何使用潜在的 "不分配" (LDA) 模型分析文本数据。LDA 模型是一个主题模型, 用于发现文档集合中的基础主题, 并在主题中推断单词概率。在准备文本数据和拟合模型时, 请使用以下步骤:

* 从网站读取 HTML 代码。
* 分析 HTML 代码并提取相关数据。
* 使用标准预处理技术准备文本数据以进行分析。
* 适合主题模型并可视化结果。

### 导入数据

使用webread函数从<https://www.aozora.gr.jp/cards/000148/files/789_14547.html>读取 "吾輩は猫である" 中的数据夏目漱石。

使用weboptions函数指定文本的字符编码。若要查找 html 的正确字符编码, 请查看 html 代码的标头。对于此文件, 指定要 "Shift\_JIS"的字符编码.

url = "https://www.aozora.gr.jp/cards/000148/files/789\_14547.html";

options = weboptions('CharacterEncoding','Shift\_JIS');

code = webread(url,options);

查看 HTML 代码的前几行。

extractBefore(code,"<script")

ans =

'<?xml version="1.0" encoding="Shift\_JIS"?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"

"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xml:lang="ja" >

<head>

<meta http-equiv="Content-Type" content="text/html;charset=Shift\_JIS" />

<meta http-equiv="content-style-type" content="text/css" />

<link rel="stylesheet" type="text/css" href="../../aozora.css" />

<title>夏目漱石 吾輩は猫である</title>

'

使用extractHTMLText从 HTML 中提取文本数据。通过newline符拆分文本。

textData = extractHTMLText(code);

textData = string(split(textData,newline));

textData(1:10)

ans = 10×1 string array

"吾輩は猫である"

""

"夏目漱石"

""

" "

""

"一"

""

" 吾輩は猫である。名前はまだ無い。"

" どこで生れたかとんと見当がつかぬ。何でも薄暗いじめじめした所でニャーニャー泣いていた事だけは記憶している。吾輩はここで始めて人間というものを見た。しかもあとで聞くとそれは書生という人間中で一番獰悪な種族であったそうだ。この書生というのは時々我々を捕えて煮て食うという話である。しかしその当時は何という考もなかったから別段恐しいとも思わなかった。ただ彼の掌に載せられてスーと持ち上げられた時何だかフワフワした感じがあったばかりである。掌の上で少し落ちついて書生の顔を見たのがいわゆる人間というものの見始であろう。この時妙なものだと思った感じが今でも残っている。第一毛をもって装飾されべきはずの顔がつるつるしてまるで薬缶だ。その後猫にもだいぶ逢ったがこんな片輪には一度も出会わした事がない。のみならず顔の真中があまりに突起している。そうしてその穴の中から時々ぷうぷうと煙を吹く。どうも咽せぽくて実に弱った。これが人間の飲む煙草というものである事はようやくこの頃知った。"

删除文本的空行。

idx = textData == "";

textData(idx) = [];

textData(1:10)

ans = 10×1 string array

"吾輩は猫である"

"夏目漱石"

" "

"一"

" 吾輩は猫である。名前はまだ無い。"

" どこで生れたかとんと見当がつかぬ。何でも薄暗いじめじめした所でニャーニャー泣いていた事だけは記憶している。吾輩はここで始めて人間というものを見た。しかもあとで聞くとそれは書生という人間中で一番獰悪な種族であったそうだ。この書生というのは時々我々を捕えて煮て食うという話である。しかしその当時は何という考もなかったから別段恐しいとも思わなかった。ただ彼の掌に載せられてスーと持ち上げられた時何だかフワフワした感じがあったばかりである。掌の上で少し落ちついて書生の顔を見たのがいわゆる人間というものの見始であろう。この時妙なものだと思った感じが今でも残っている。第一毛をもって装飾されべきはずの顔がつるつるしてまるで薬缶だ。その後猫にもだいぶ逢ったがこんな片輪には一度も出会わした事がない。のみならず顔の真中があまりに突起している。そうしてその穴の中から時々ぷうぷうと煙を吹く。どうも咽せぽくて実に弱った。これが人間の飲む煙草というものである事はようやくこの頃知った。"

" この書生の掌の裏でしばらくはよい心持に坐っておったが、しばらくすると非常な速力で運転し始めた。書生が動くのか自分だけが動くのか分らないが無暗に眼が廻る。胸が悪くなる。到底助からないと思っていると、どさりと音がして眼から火が出た。それまでは記憶しているがあとは何の事やらいくら考え出そうとしても分らない。"

" ふと気が付いて見ると書生はいない。たくさんおった兄弟が一疋も見えぬ。肝心の母親さえ姿を隠してしまった。その上今までの所とは違って無暗に明るい。眼を明いていられぬくらいだ。はてな何でも容子がおかしいと、のそのそ這い出して見ると非常に痛い。吾輩は藁の上から急に笹原の中へ棄てられたのである。"

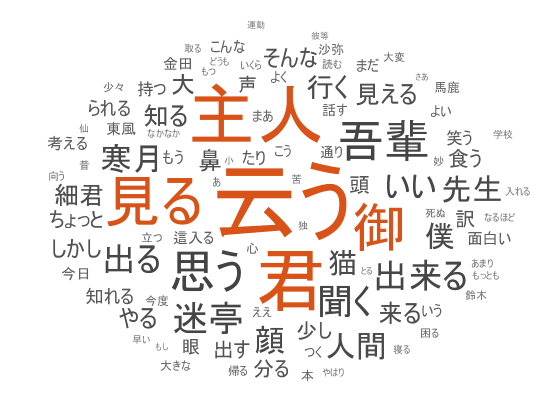
" ようやくの思いで笹原を這い出すと向うに大きな池がある。吾輩は池の前に坐ってどうしたらよかろうと考えて見た。別にこれという分別も出ない。しばらくして泣いたら書生がまた迎に来てくれるかと考え付いた。ニャー、ニャーと試みにやって見たが誰も来ない。そのうち池の上をさらさらと風が渡って日が暮れかかる。腹が非常に減って来た。泣きたくても声が出ない。仕方がない、何でもよいから食物のある所まであるこうと決心をしてそろりそろりと池を左りに廻り始めた。どうも非常に苦しい。そこを我慢して無理やりに這って行くとようやくの事で何となく人間臭い所へ出た。ここへ這入ったら、どうにかなると思って竹垣の崩れた穴から、とある邸内にもぐり込んだ。縁は不思議なもので、もしこの竹垣が破れていなかったなら、吾輩はついに路傍に餓死したかも知れんのである。一樹の蔭とはよく云ったものだ。この垣根の穴は今日に至るまで吾輩が隣家の三毛を訪問する時の通路になっている。さて邸へは忍び込んだもののこれから先どうして善いか分らない。そのうちに暗くなる、腹は減る、寒さは寒し、雨が降って来るという始末でもう一刻の猶予が出来なくなった。仕方がないからとにかく明るくて暖かそうな方へ方へとあるいて行く。今から考えるとその時はすでに家の内に這入っておったのだ。ここで吾輩は彼の書生以外の人間を再び見るべき機会に遭遇したのである。第一に逢ったのがおさんである。これは前の書生より一層乱暴な方で吾輩を見るや否やいきなり頸筋をつかんで表へ抛り出した。いやこれは駄目だと思ったから眼をねぶって運を天に任せていた。しかしひもじいのと寒いのにはどうしても我慢が出来ん。吾輩は再びおさんの隙を見て台所へ這い上った。すると間もなくまた投げ出された。吾輩は投げ出されては這い上り、這い上っては投げ出され、何でも同じ事を四五遍繰り返したのを記憶している。その時におさんと云う者はつくづくいやになった。この間おさんの三馬を偸んでこの返報をしてやってから、やっと胸の痞が下りた。吾輩が最後につまみ出されようとしたときに、この家の主人が騒々しい何だといいながら出て来た。下女は吾輩をぶら下げて主人の方へ向けてこの宿なしの小猫がいくら出しても出しても御台所へ上って来て困りますという。主人は鼻の下の黒い毛を撚りながら吾輩の顔をしばらく眺めておったが、やがてそんなら内へ置いてやれといったまま奥へ這入ってしまった。主人はあまり口を聞かぬ人と見えた。下女は口惜しそうに吾輩を台所へ抛り出した。かくして吾輩はついにこの家を自分の住家と極める事にしたのである。"

" 吾輩の主人は滅多に吾輩と顔を合せる事がない。職業は教師だそうだ。学校から帰ると終日書斎に這入ったぎりほとんど出て来る事がない。家のものは大変な勉強家だと思っている。当人も勉強家であるかのごとく見せている。しかし実際はうちのものがいうような勤勉家ではない。吾輩は時々忍び足に彼の書斎を覗いて見るが、彼はよく昼寝をしている事がある。時々読みかけてある本の上に涎をたらしている。彼は胃弱で皮膚の色が淡黄色を帯びて弾力のない不活溌な徴候をあらわしている。その癖に大飯を食う。大飯を食った後でタカジヤスターゼを飲む。飲んだ後で書物をひろげる。二三ページ読むと眠くなる。涎を本の上へ垂らす。これが彼の毎夜繰り返す日課である。吾輩は猫ながら時々考える事がある。教師というものは実に楽なものだ。人間と生れたら教師となるに限る。こんなに寝ていて勤まるものなら猫にでも出来ぬ事はないと。それでも主人に云わせると教師ほどつらいものはないそうで彼は友達が来る度に何とかかんとか不平を鳴らしている。"

在 word 云中可视化文本数据。

figure

wordcloud(textData);



### 准备用于分析的文本数据

使用tokenizedDocument标记化文本并查看前几个文档。

documents = tokenizedDocument(textData);

documentsRaw = documents;

documents(1:10)

ans =

10×1 tokenizedDocument:

5 tokens: 吾輩 は 猫 で ある

2 tokens: 夏目 漱石

0 tokens:

1 tokens: 一

11 tokens: 吾輩 は 猫 で ある 。 名前 は まだ 無い 。

264 tokens: どこ で 生れ た か とんと 見当 が つか ぬ 。 何 でも 薄暗い じめじめ し た 所 で ニャーニャー 泣い て いた事 だけ は 記憶 し て いる 。 吾輩 は ここ で 始め て 人間 という もの を 見 た 。 しかも あと で 聞く と それ は 書生 という 人間 中 で 一番 獰悪 な 種族 で あっ た そう だ 。 この 書生 という の は 時々 我々 を 捕え て 煮 て 食う という 話 で ある 。 しかし その 当時 は 何 という 考 も なかっ た から 別段 恐し いとも 思わ なかっ た 。 ただ 彼 の 掌 に 載せ られ て スー と 持ち上げ られ た 時 何だか フワフワ し た 感じ が あっ た ばかり で ある 。 掌 の 上 で 少し 落ちつい て 書生 の 顔 を 見 た の が いわゆる 人間 という もの の 見 始 で あろ う 。 この 時 妙 な もの だ と 思っ た 感じ が 今 でも 残っ て いる 。 第 一 毛 をもって 装飾 さ れ べき はず の 顔 が つるつる し て まるで 薬缶 だ 。 その後 猫 に も だいぶ 逢っ た が こんな 片 輪 に は 一 度 も 出会わ し た 事 が ない 。 のみ なら ず 顔 の 真中 が あまりに 突起 し て いる 。 そうして その 穴 の 中 から 時々 ぷうぷうと 煙 を 吹く 。 どうも 咽 せ ぽく て 実に 弱っ た 。 これ が 人間 の 飲む 煙草 という もの で ある 事 は ようやく この 頃 知っ た 。

100 tokens: この 書生 の 掌 の 裏 で しばらく は よい 心持 に 坐っ て おっ た が 、 しばらく する と 非常 な 速力 で 運転 し 始め た 。 書生 が 動く の か 自分 だけ が 動く の か 分ら ない が 無 暗に 眼 が 廻る 。 胸 が 悪く なる 。 到底 助から ない と 思っ て いる と 、 ど さり と 音 が し て 眼 から 火 が 出 た 。 それ まで は 記憶 し て いる が あと は 何 の 事 やら いくら 考え出そ う として も 分ら ない 。

92 tokens: ふと 気 が 付い て 見る と 書生 は い ない 。 たくさん おっ た 兄弟 が 一疋 も 見え ぬ 。 肝心 の 母親 さえ 姿 を 隠し て しまっ た 。 その 上今 まで の 所 と は 違っ て 無 暗に 明るい 。 眼 を 明い て い られ ぬ くらい だ 。 はてな 何 でも 容子 が おかしい と 、 のそのそ 這い出し て 見る と 非常 に 痛い 。 吾輩 は 藁 の 上 から 急 に 笹原 の 中 へ 棄て られ た の で ある 。

693 tokens: ようやく の 思い で 笹原 を 這い出す と 向う に 大きな 池 が ある 。 吾輩 は 池 の 前 に 坐っ て どう し たら よかろ う と 考え て 見 た 。 別に これ という 分別 も 出 ない 。 しばらく し て 泣い たら 書生 が また 迎 に 来 て くれる か と 考え付い た 。 ニャー 、 ニャー と 試み に やっ て 見 た が 誰 も 来 ない 。 そのうち 池 の 上 を さらさら と 風 が 渡っ て 日 が 暮れ かかる 。 腹 が 非常 に 減っ て 来 た 。 泣き たく て も 声 が 出 ない 。 仕方 が ない 、 何 で も よい から 食物 の ある 所 まで ある こう と 決心 を し て そろ り そろりと 池 を 左 り に 廻り 始め た 。 どうも 非常 に 苦しい 。 そこ を 我慢 し て 無理やり に 這っ て 行く と ようやく の 事 で 何となく 人間 臭い 所 へ 出 た 。 ここ へ 這入っ たら 、 どうにか なる と 思っ て 竹垣 の 崩れ た 穴 から 、 とある 邸 内 に もぐり込ん だ 。 縁 は 不思議 な もの で 、 もし この 竹垣 が 破れ て い なかっ た なら 、 吾輩 は ついに 路傍 に 餓死 し た かも 知れ ん の で ある 。 一樹 の 蔭 と は よく 云っ た もの だ 。 この 垣根 の 穴 は 今日 に 至る まで 吾輩 が 隣家 の 三 毛 を 訪問 する 時 の 通路 に なっ て いる 。 さて 邸 へ は 忍び込ん だ ものの これから 先 どうして 善い か 分ら ない 。 その うち に 暗く なる 、 腹 は 減る 、 寒 さ は 寒し 、 雨 が 降っ て 来る という 始末 で もう 一刻 の 猶予 が 出来 なく なっ た 。 仕方 が ない から とにかく 明るく て 暖か そう な 方 へ 方 へ と あるい て 行く 。 今 から 考える と その 時 は すでに 家 の 内 に 這入っ て おっ た の だ 。 ここ で 吾輩 は 彼 の 書生 以外 の 人間 を 再び 見る べき 機会 に 遭遇 し た の で ある 。 第 一 に 逢っ た の が お さん で ある 。 こ…

276 tokens: 吾輩 の 主人 は 滅多 に 吾輩 と 顔 を 合せる 事 が ない 。 職業 は 教師 だ そう だ 。 学校 から 帰る と 終日 書斎 に 這入っ た ぎりほとんど 出 て 来る 事 が ない 。 家 の もの は 大変 な 勉強 家 だ と 思っ て いる 。 当人 も 勉強 家 で ある か の ごとく 見せ て いる 。 しかし 実際 は うち の もの が いう よう な 勤勉 家 で は ない 。 吾輩 は 時々 忍び足 に 彼 の 書斎 を 覗い て 見る が 、 彼 は よく 昼寝 を し て いる 事 が ある 。 時々 読みかけ て ある 本 の 上 に 涎 を たらし て いる 。 彼 は 胃弱 で 皮膚 の 色 が 淡 黄色 を 帯び て 弾力 の ない 不 活溌 な 徴候 を あらわし て いる 。 その 癖 に 大飯 を 食う 。 大飯 を 食っ た 後 で タカジヤスターゼ を 飲む 。 飲ん だ 後 で 書物 を ひろげる 。 二 三 ページ 読む と 眠く なる 。 涎 を 本 の 上 へ 垂らす 。 これ が 彼 の 毎夜 繰り返す 日課 で ある 。 吾輩 は 猫 ながら 時々 考える 事 が ある 。 教師 という もの は 実に 楽 な もの だ 。 人間 と 生れ たら 教師 と なる に 限る 。 こんなに 寝 て い て 勤まる もの なら 猫 に でも 出来 ぬ 事 は ない と 。 それでも 主人 に 云わ せる と 教師 ほど つらい もの は ない そう で 彼 は 友達 が 来る 度 に 何とか かん とか 不平 を 鳴らし て いる 。

删除停止单词。

documents = removeStopWords(documents);

documents(1:10)

ans =

10×1 tokenizedDocument:

2 tokens: 吾輩 猫

2 tokens: 夏目 漱石

0 tokens:

0 tokens:

6 tokens: 吾輩 猫 。 まだ 無い 。

117 tokens: 生れ とんと 見当 つか ぬ 。 薄暗い じめじめ ニャーニャー 泣い いた事 記憶 。 吾輩 始め 人間 という 。 しかも 聞く 書生 という 人間 一番 獰悪 種族 あっ 。 書生 という 時々 捕え 煮 食う という 。 しかし 当時 という 考 なかっ 別段 恐し いとも 思わ なかっ 。 掌 載せ られ スー 持ち上げ られ 何だか フワフワ あっ 。 掌 少し 落ちつい 書生 顔 いわゆる 人間 という 始 あろ 。 妙 思っ 残っ 。 毛 をもって 装飾 れ べき 顔 つるつる まるで 薬缶 。 猫 だいぶ 逢っ こんな 片 出会わ 。 のみ なら 顔 真中 あまりに 突起 。 そうして 穴 時々 ぷうぷうと 煙 吹く 。 どうも 咽 せ ぽく 実に 弱っ 。 人間 飲む 煙草 という ようやく 知っ 。

43 tokens: 書生 掌 裏 しばらく よい 心持 坐っ おっ 、 しばらく 非常 速力 運転 始め 。 書生 動く 動く 分ら 無 暗に 眼 廻る 。 胸 悪く 。 到底 助から 思っ 、 ど さり 音 眼 出 。 記憶 いくら 考え出そ として 分ら 。

46 tokens: ふと 付い 見る 書生 。 おっ 兄弟 一疋 見え ぬ 。 肝心 母親 姿 隠し しまっ 。 上今 違っ 無 暗に 明るい 。 眼 明い られ ぬ くらい 。 はてな 容子 おかしい 、 のそのそ 這い出し 見る 非常 痛い 。 吾輩 藁 急 笹原 棄て られ 。

323 tokens: ようやく 思い 笹原 這い出す 向う 大きな 池 。 吾輩 池 坐っ たら よかろ 考え 。 別に という 分別 出 。 しばらく 泣い たら 書生 迎 来 くれる 考え付い 。 ニャー 、 ニャー 試み やっ 来 。 そのうち 池 さらさら 風 渡っ 暮れ かかる 。 腹 非常 減っ 来 。 泣き たく 声 出 。 仕方 、 よい 食物 こう 決心 そろ り そろりと 池 り 廻り 始め 。 どうも 非常 苦しい 。 我慢 無理やり 這っ 行く ようやく 何となく 人間 臭い 出 。 這入っ たら 、 どうにか 思っ 竹垣 崩れ 穴 、 とある 邸 もぐり込ん 。 縁 不思議 、 もし 竹垣 破れ なかっ なら 、 吾輩 ついに 路傍 餓死 かも 知れ 。 一樹 蔭 よく 云っ 。 垣根 穴 今日 至る 吾輩 隣家 毛 訪問 通路 なっ 。 さて 邸 忍び込ん ものの これから どうして 善い 分ら 。 暗く 、 腹 減る 、 寒 寒し 、 雨 降っ 来る という 始末 もう 一刻 猶予 出来 なく なっ 。 仕方 とにかく 明るく 暖か あるい 行く 。 考える すでに 這入っ おっ 。 吾輩 書生 以外 人間 再び 見る べき 機会 遭遇 。 逢っ 。 書生 一層 乱暴 吾輩 見る 否や いきなり 頸筋 つかん 表 抛り出し 。 駄目 思っ 眼 ねぶっ 運 天 任せ 。 しかし ひもじい 寒い どうしても 我慢 出来 。 吾輩 再び おさん 隙 台所 這い 上っ 。 すると 間もなく 投げ出さ れ 。 吾輩 投げ出さ れ 這い 上り 、 這い 上っ 投げ出さ れ 、 遍 繰り返し 記憶 。 おさん 云う つくづく なっ 。 この間 おさん 馬 偸 んで 返報 やっ 、 やっと 胸 痞 下り 。 吾輩 最後 つまみ出さ れよ 、 主人 騒々しい いい ながら 出 来 。 下女 吾輩 ぶら下げ 主人 向け 宿 なし 小 猫 いくら 出し 出し 御台 上っ 来 困り いう 。 主人 鼻 黒い 毛 撚り ながら 吾輩 顔 しばらく 眺め おっ 、 やがて そん なら 置い やれ といった 奥 這入っ しまっ 。 主人 あまり 聞か ぬ 見え 。 下女 口惜し 吾輩 台所 抛り出し 。 かくして 吾輩 ついに 住 極める 。

122 tokens: 吾輩 主人 滅多 吾輩 顔 合せる 。 職業 教師 。 学校 帰る 終日 書斎 這入っ ぎりほとんど 出 来る 。 大変 勉強 思っ 。 当人 勉強 ごとく 見せ 。 しかし 実際 いう 勤勉 。 吾輩 時々 忍び足 書斎 覗い 見る 、 よく 昼寝 。 時々 読みかけ 本 涎 たらし 。 胃弱 皮膚 色 淡 黄色 帯び 弾力 不 活溌 徴候 あらわし 。 癖 大飯 食う 。 大飯 食っ タカジヤスターゼ 飲む 。 飲ん 書物 ひろげる 。 ページ 読む 眠く 。 涎 本 垂らす 。 毎夜 繰り返す 日課 。 吾輩 猫 ながら 時々 考える 。 教師 という 実に 。 人間 生れ たら 教師 限る 。 こんなに 寝 勤まる なら 猫 出来 ぬ 。 それでも 主人 云わ 教師 つらい 友達 来る 何とか かん 不平 鳴らし 。

擦除标点符号。

documents = erasePunctuation(documents);

documents(1:10)

ans =

10×1 tokenizedDocument:

2 tokens: 吾輩 猫

2 tokens: 夏目 漱石

0 tokens:

0 tokens:

4 tokens: 吾輩 猫 まだ 無い

102 tokens: 生れ とんと 見当 つか ぬ 薄暗い じめじめ ニャーニャー 泣い いた事 記憶 吾輩 始め 人間 という しかも 聞く 書生 という 人間 一番 獰悪 種族 あっ 書生 という 時々 捕え 煮 食う という しかし 当時 という 考 なかっ 別段 恐し いとも 思わ なかっ 掌 載せ られ スー 持ち上げ られ 何だか フワフワ あっ 掌 少し 落ちつい 書生 顔 いわゆる 人間 という 始 あろ 妙 思っ 残っ 毛 をもって 装飾 れ べき 顔 つるつる まるで 薬缶 猫 だいぶ 逢っ こんな 片 出会わ のみ なら 顔 真中 あまりに 突起 そうして 穴 時々 ぷうぷうと 煙 吹く どうも 咽 せ ぽく 実に 弱っ 人間 飲む 煙草 という ようやく 知っ

36 tokens: 書生 掌 裏 しばらく よい 心持 坐っ おっ しばらく 非常 速力 運転 始め 書生 動く 動く 分ら 無 暗に 眼 廻る 胸 悪く 到底 助から 思っ ど さり 音 眼 出 記憶 いくら 考え出そ として 分ら

38 tokens: ふと 付い 見る 書生 おっ 兄弟 一疋 見え ぬ 肝心 母親 姿 隠し しまっ 上今 違っ 無 暗に 明るい 眼 明い られ ぬ くらい はてな 容子 おかしい のそのそ 這い出し 見る 非常 痛い 吾輩 藁 急 笹原 棄て られ

274 tokens: ようやく 思い 笹原 這い出す 向う 大きな 池 吾輩 池 坐っ たら よかろ 考え 別に という 分別 出 しばらく 泣い たら 書生 迎 来 くれる 考え付い ニャー ニャー 試み やっ 来 そのうち 池 さらさら 風 渡っ 暮れ かかる 腹 非常 減っ 来 泣き たく 声 出 仕方 よい 食物 こう 決心 そろ り そろりと 池 り 廻り 始め どうも 非常 苦しい 我慢 無理やり 這っ 行く ようやく 何となく 人間 臭い 出 這入っ たら どうにか 思っ 竹垣 崩れ 穴 とある 邸 もぐり込ん 縁 不思議 もし 竹垣 破れ なかっ なら 吾輩 ついに 路傍 餓死 かも 知れ 一樹 蔭 よく 云っ 垣根 穴 今日 至る 吾輩 隣家 毛 訪問 通路 なっ さて 邸 忍び込ん ものの これから どうして 善い 分ら 暗く 腹 減る 寒 寒し 雨 降っ 来る という 始末 もう 一刻 猶予 出来 なく なっ 仕方 とにかく 明るく 暖か あるい 行く 考える すでに 這入っ おっ 吾輩 書生 以外 人間 再び 見る べき 機会 遭遇 逢っ 書生 一層 乱暴 吾輩 見る 否や いきなり 頸筋 つかん 表 抛り出し 駄目 思っ 眼 ねぶっ 運 天 任せ しかし ひもじい 寒い どうしても 我慢 出来 吾輩 再び おさん 隙 台所 這い 上っ すると 間もなく 投げ出さ れ 吾輩 投げ出さ れ 這い 上り 這い 上っ 投げ出さ れ 遍 繰り返し 記憶 おさん 云う つくづく なっ この間 おさん 馬 偸 んで 返報 やっ やっと 胸 痞 下り 吾輩 最後 つまみ出さ れよ 主人 騒々しい いい ながら 出 来 下女 吾輩 ぶら下げ 主人 向け 宿 なし 小 猫 いくら 出し 出し 御台 上っ 来 困り いう 主人 鼻 黒い 毛 撚り ながら 吾輩 顔 しばらく 眺め おっ やがて そん なら 置い やれ といった 奥 這入っ しまっ 主人 あまり 聞か ぬ 見え 下女 口惜し 吾輩 台所 抛り出し かくして 吾輩 ついに 住 極める

101 tokens: 吾輩 主人 滅多 吾輩 顔 合せる 職業 教師 学校 帰る 終日 書斎 這入っ ぎりほとんど 出 来る 大変 勉強 思っ 当人 勉強 ごとく 見せ しかし 実際 いう 勤勉 吾輩 時々 忍び足 書斎 覗い 見る よく 昼寝 時々 読みかけ 本 涎 たらし 胃弱 皮膚 色 淡 黄色 帯び 弾力 不 活溌 徴候 あらわし 癖 大飯 食う 大飯 食っ タカジヤスターゼ 飲む 飲ん 書物 ひろげる ページ 読む 眠く 涎 本 垂らす 毎夜 繰り返す 日課 吾輩 猫 ながら 時々 考える 教師 という 実に 人間 生れ たら 教師 限る こんなに 寝 勤まる なら 猫 出来 ぬ それでも 主人 云わ 教師 つらい 友達 来る 何とか かん 不平 鳴らし

使用normalizeWords Lemmatize 文本.

documents = normalizeWords(documents);

documents(1:10)

ans =

10×1 tokenizedDocument:

2 tokens: 吾輩 猫

2 tokens: 夏目 漱石

0 tokens:

0 tokens:

4 tokens: 吾輩 猫 まだ 無い

102 tokens: 生れる とんと 見当 つく ぬ 薄暗い じめじめ ニャーニャー 泣く いた事 記憶 吾輩 始める 人間 という しかも 聞く 書生 という 人間 一番 獰悪 種族 ある 書生 という 時々 捕える 煮る 食う という しかし 当時 という 考 ない 別段 恐い いとも 思う ない 掌 載せる られる スー 持ち上げる られる 何だか フワフワ ある 掌 少し 落ちつく 書生 顔 いわゆる 人間 という 始 ある 妙 思う 残る 毛 をもって 装飾 れる べし 顔 つるつる まるで 薬缶 猫 だいぶ 逢う こんな 片 出会う のみ なる 顔 真中 あまりに 突起 そうして 穴 時々 ぷうぷうと 煙 吹く どうも 咽 する ぽい 実に 弱る 人間 飲む 煙草 という ようやく 知る

36 tokens: 書生 掌 裏 しばらく よい 心持 坐る おる しばらく 非常 速力 運転 始める 書生 動く 動く 分る 無 暗に 眼 廻る 胸 悪い 到底 助かる 思う ど さる 音 眼 出る 記憶 いくら 考え出す として 分る

38 tokens: ふと 付く 見る 書生 おる 兄弟 一疋 見える ぬ 肝心 母親 姿 隠す しまう 上今 違う 無 暗に 明るい 眼 明く られる ぬ くらい はてな 容子 おかしい のそのそ 這い出す 見る 非常 痛い 吾輩 藁 急 笹原 棄てる られる

274 tokens: ようやく 思い 笹原 這い出す 向う 大きな 池 吾輩 池 坐る た よい 考える 別に という 分別 出る しばらく 泣く た 書生 迎 来る くれる 考え付く ニャー ニャー 試み やる 来る そのうち 池 さらさら 風 渡る 暮れ かかる 腹 非常 減る 来る 泣く たい 声 出る 仕方 よい 食物 こう 決心 そる り そろりと 池 り 廻る 始める どうも 非常 苦しい 我慢 無理やり 這う 行く ようやく 何となく 人間 臭い 出る 這入る た どうにか 思う 竹垣 崩れる 穴 とある 邸 もぐり込む 縁 不思議 もし 竹垣 破れる ない だ 吾輩 ついに 路傍 餓死 かも 知れる 一樹 蔭 よく 云う 垣根 穴 今日 至る 吾輩 隣家 毛 訪問 通路 なる さて 邸 忍び込む ものの これから どうして 善い 分る 暗い 腹 減る 寒い 寒い 雨 降る 来る という 始末 もう 一刻 猶予 出来る ない なる 仕方 とにかく 明るい 暖かい あるく 行く 考える すでに 這入る おる 吾輩 書生 以外 人間 再び 見る べし 機会 遭遇 逢う 書生 一層 乱暴 吾輩 見る 否や いきなり 頸筋 つかむ 表 抛り出す 駄目 思う 眼 ねぶる 運 天 任せる しかし ひもじい 寒い どうしても 我慢 出来る 吾輩 再び おさん 隙 台所 這う 上る すると 間もなく 投げ出す れる 吾輩 投げ出す れる 這う 上る 這う 上る 投げ出す れる 遍 繰り返す 記憶 おさん 云う つくづく なる この間 おさん 馬 偸 んで 返報 やる やっと 胸 痞 下りる 吾輩 最後 つまみ出す れる 主人 騒々しい いう ながら 出る 来る 下女 吾輩 ぶら下げる 主人 向ける 宿 ない 小 猫 いくら 出す 出す 御台 上る 来る 困る いう 主人 鼻 黒い 毛 撚る ながら 吾輩 顔 しばらく 眺める おる やがて そん だ 置く やる といった 奥 這入る しまう 主人 あまり 聞く ぬ 見える 下女 口惜しい 吾輩 台所 抛り出す かくして 吾輩 ついに 住 極める

101 tokens: 吾輩 主人 滅多 吾輩 顔 合せる 職業 教師 学校 帰る 終日 書斎 這入る ぎりほとんど 出る 来る 大変 勉強 思う 当人 勉強 ごとし 見せる しかし 実際 いう 勤勉 吾輩 時々 忍び足 書斎 覗く 見る よく 昼寝 時々 読みかける 本 涎 たらす 胃弱 皮膚 色 淡い 黄色 帯びる 弾力 不 活溌 徴候 あらわす 癖 大飯 食う 大飯 食う タカジヤスターゼ 飲む 飲む 書物 ひろげる ページ 読む 眠い 涎 本 垂らす 毎夜 繰り返す 日課 吾輩 猫 ながら 時々 考える 教師 という 実に 人間 生れる た 教師 限る こんなに 寝る 勤まる だ 猫 出来る ぬ それでも 主人 云う 教師 つらい 友達 来る 何とか かん 不平 鳴らす

某些预处理步骤 (如删除停止字和擦除标点) 返回空文档。使用removeEmptyDocuments函数删除空文档。

documents = removeEmptyDocuments(documents);

### 创建预处理功能

创建执行预处理的函数对于以相同的方式准备不同的文本数据集合非常有用。例如, 可以使用函数来使用与定型数据相同的步骤来预处理新数据。

创建先标记和预处理用于分析的文本数据的函数。函数preprocessJapaneseText, 执行以下步骤:

1. 使用tokenizedDocument标记化文本.
2. 使用erasePunctuation擦除标点.
3. 使用removeStopWords删除停止字词列表 (如 "あそこ"、"あたり" 和 "あちら").
4. 用normalizeWords Lemmatize 单词.

使用removeEmptyDocuments函数预处理后删除空文档。使用预处理函数后删除文档可以更轻松地删除相应的数据, 如其他来源的标签。

在此示例中, 使用在示例末尾列出的预处理函数preprocessJapaneseText准备文本数据。

documents = preprocessJapaneseText(textData);

documents(1:5)

ans =

5×1 tokenizedDocument:

2 tokens: 吾輩 猫

2 tokens: 夏目 漱石

0 tokens:

0 tokens:

4 tokens: 吾輩 猫 まだ 無い

删除空文档。

documents = removeEmptyDocuments(documents);

### 获取部分语音标记

获取令牌详细信息, 然后查看前几个标记的详细信息。

tdetails = tokenDetails(documents);

head(tdetails)

ans=8×7 table

Token DocumentNumber LineNumber Type Language PartOfSpeech Lemma

\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_

"吾輩" 1 1 letters ja pronoun "吾輩"

"猫" 1 1 letters ja noun "猫"

"夏目" 2 1 letters ja proper-noun "夏目"

"漱石" 2 1 letters ja proper-noun "漱石"

"吾輩" 3 1 letters ja pronoun "吾輩"

"猫" 3 1 letters ja noun "猫"

"まだ" 3 1 letters ja adverb "まだ"

"無い" 3 1 letters ja adjective "無い"

表中的PartOfSpeech变量包含令牌的词性标记。分别创建所有名词和形容词的字云。

figure

idx = tdetails.PartOfSpeech == "noun";

tokens = tdetails.Token(idx);

subplot(1,2,1)

wordcloud(tokens);

title("Nouns")

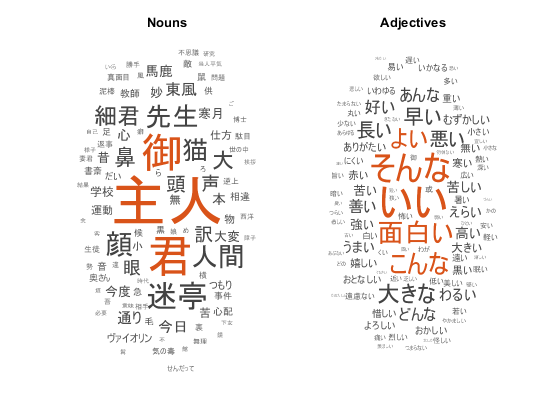
idx = tdetails.PartOfSpeech == "adjective";

tokens = tdetails.Token(idx);

subplot(1,2,2)

wordcloud(tokens);

title("Adjectives")



### 拟合主题模型

将潜在的可分配 (LDA) 主题模型拟合到数据中。LDA 模型在文档集合中发现基础主题, 并在主题中推断单词概率。

要将 LDA 模型适合数据, 首先必须创建一个单词袋模型。单词袋模型 (也称为术语频率计数器) 记录单词在集合的每个文档中出现的次数。使用bagOfWords创建一个词袋模型.

bag = bagOfWords(documents);

从单词袋模型中删除空文档。

bag = removeEmptyDocuments(bag);

使用fitlda为七个主题拟合 LDA 模型。要抑制详细输出, 请将'Verbose'设置为0.

numTopics = 7;

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

使用 word 云可视化前四个主题。

figure

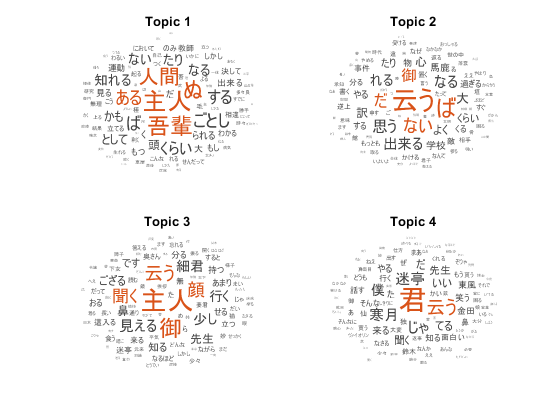
for i = 1:4

subplot(2,2,i)

wordcloud(mdl,i);

title("Topic " + i)

end



使用堆积条形图可视化多个主题混合物。随机查看五输入文档, 并可视化相应的主题混合物。

numDocuments = numel(documents);

idx = randperm(numDocuments,5);

documents(idx)

ans =

5×1 tokenizedDocument:

10 tokens: 馬鹿馬鹿しい じゃ いくら 唐津 掘る 来る たって 山の芋 銭 たまる

6 tokens: ええ 少し 御 話す する 思う

26 tokens: しかし 娘 寒月 来る 主人 鈴木 君 聞く 通り 述べる 鈴木 君 迷惑 云う 顔 付 しきりに 主人 くばる せる 主人 不導体 ごとし 一向 電気 感染

8 tokens: 御 母 あさまい 夕べ 泥棒 這入る 姉 尋ねる

5 tokens: 真面目 現に 御 友達 手紙

topicMixtures = transform(mdl,documents(idx));

figure

barh(topicMixtures(1:5,:),'stacked')

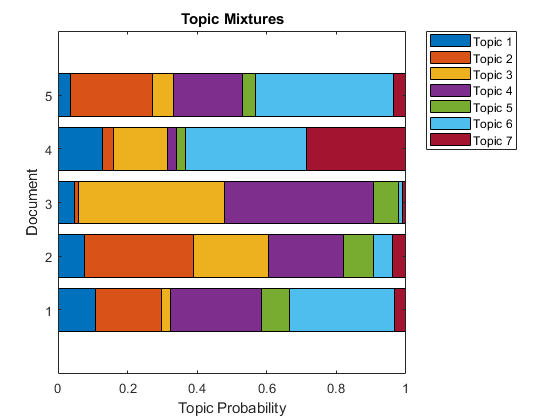
xlim([0 1])

title("Topic Mixtures")

xlabel("Topic Probability")

ylabel("Document")

legend("Topic " + string(1:numTopics),'Location','northeastoutside')



**预处理函数示例**

函数preprocessJapaneseText, 执行以下步骤:

1. 使用tokenizedDocument标记化文本.
2. 使用erasePunctuation擦除标点.
3. 使用removeStopWords删除停止字词列表 (如 "あそこ"、"あたり" 和 "あちら").
4. 用normalizeWords Lemmatize 单词.

function documents = preprocessJapaneseText(textData)

% Tokenize the text.

documents = tokenizedDocument(textData);

% Erase the punctuation.

documents = erasePunctuation(documents);

% Remove a list of stop words.

documents = removeStopWords(documents);

% Lemmatize the words.

documents = normalizeWords(documents,'Style','lemma');

end

# 建模与预测

## 使用深度学习对文本数据进行分类

此示例使用:

* [深度学习工具箱](https://www.mathworks.com/help/deeplearning/index.html)
* [文本分析工具箱](https://www.mathworks.com/help/textanalytics/index.html)

此示例演示如何使用深度学习长短期内存 (LSTM) 网络对天气报告的文本描述进行分类。

文本数据自然是连续的。一段文本是一个单词序列, 它们之间可能存在依赖关系。要学习和使用长期依赖关系来分类序列数据, 请使用 LSTM 神经网络。LSTM 网络是一种周期性神经网络 (RNN), 可以学习序列数据的时间步长之间的长期依赖关系。

要将文本输入到 LSTM 网络, 请首先将文本数据转换为数字序列。您可以使用将文档映射到数字索引序列的文字编码来实现此目的。为了获得更好的结果, 还要在网络中包含一个字嵌入层。Word 嵌入将词汇中的单词映射为数字向量, 而不是标量索引。这些嵌入捕获单词的语义细节, 因此具有相似含义的词具有相似的向量。它们还通过向量算法对单词之间的关系进行建模。例如, "国王是女王, 男人是女人" 的关系由等式国王 - 男人 + 女人 = 皇后.

在本例中, 培训和使用 LSTM 网络有四步:

* 导入和预处理数据。
* 使用单词编码将单词转换为数字序列。
* 创建和训练一个 LSTM 网络与一个字嵌入层。
* 使用训练有素的 LSTM 网络对新文本数据进行分类。

### 导入数据

导入天气报告数据。此数据包含对天气事件的标记的文本描述。若要将文本数据导入为字符串, 请将文本类型指定为'string'.

filename = "weatherReports.csv";

data = readtable(filename,'TextType','string');

head(data)

ans=8×16 table

Time event\_id state event\_type damage\_property damage\_crops begin\_lat begin\_lon end\_lat end\_lon event\_narrative storm\_duration begin\_day end\_day year end\_timestamp

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

22-Jul-2016 16:10:00 6.4433e+05 "MISSISSIPPI" "Thunderstorm Wind" "" "0.00K" 34.14 -88.63 34.122 -88.626 "Large tree down between Plantersville and Nettleton." 00:05:00 22 22 2016 22-Jul-0016 16:15:00

15-Jul-2016 17:15:00 6.5182e+05 "SOUTH CAROLINA" "Heavy Rain" "2.00K" "0.00K" 34.94 -81.03 34.94 -81.03 "One to two feet of deep standing water developed on a street on the Winthrop University campus after more than an inch of rain fell in less than an hour. One vehicle was stalled in the water." 00:00:00 15 15 2016 15-Jul-0016 17:15:00

15-Jul-2016 17:25:00 6.5183e+05 "SOUTH CAROLINA" "Thunderstorm Wind" "0.00K" "0.00K" 35.01 -80.93 35.01 -80.93 "NWS Columbia relayed a report of trees blown down along Tom Hall St." 00:00:00 15 15 2016 15-Jul-0016 17:25:00

16-Jul-2016 12:46:00 6.5183e+05 "NORTH CAROLINA" "Thunderstorm Wind" "0.00K" "0.00K" 35.64 -82.14 35.64 -82.14 "Media reported two trees blown down along I-40 in the Old Fort area." 00:00:00 16 16 2016 16-Jul-0016 12:46:00

15-Jul-2016 14:28:00 6.4332e+05 "MISSOURI" "Hail" "" "" 36.45 -89.97 36.45 -89.97 "" 00:07:00 15 15 2016 15-Jul-0016 14:35:00

15-Jul-2016 16:31:00 6.4332e+05 "ARKANSAS" "Thunderstorm Wind" "" "0.00K" 35.85 -90.1 35.838 -90.087 "A few tree limbs greater than 6 inches down on HWY 18 in Roseland." 00:09:00 15 15 2016 15-Jul-0016 16:40:00

15-Jul-2016 16:03:00 6.4343e+05 "TENNESSEE" "Thunderstorm Wind" "20.00K" "0.00K" 35.056 -89.937 35.05 -89.904 "Awning blown off a building on Lamar Avenue. Multiple trees down near the intersection of Winchester and Perkins." 00:07:00 15 15 2016 15-Jul-0016 16:10:00

15-Jul-2016 17:27:00 6.4344e+05 "TENNESSEE" "Hail" "" "" 35.385 -89.78 35.385 -89.78 "Quarter size hail near Rosemark." 00:05:00 15 15 2016 15-Jul-0016 17:32:00

删除包含空报表的表行。

idxEmpty = strlength(data.event\_narrative) == 0;

data(idxEmpty,:) = [];

本示例的目标是按event\_type列中的标签对事件进行分类。若要将数据划分为类, 请将这些标签转换为分类。

data.event\_type = categorical(data.event\_type);

使用直方图查看数据中类的分布情况。要使标签更易于阅读, 请增加图形的宽度。

f = figure;

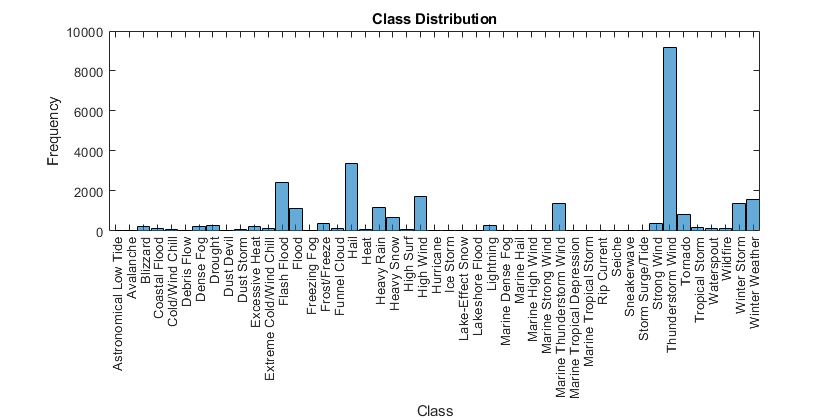
f.Position(3) = 1.5\*f.Position(3);

h = histogram(data.event\_type);

xlabel("Class")

ylabel("Frequency")

title("Class Distribution")



数据的类不平衡, 许多类包含很少的观察值。当类以这种方式不平衡时, 网络可能收敛到一个不太准确的模型。若要防止此问题, 请删除出现少于十次的任何类。

从直方图中获取类和类名的频率计数。

classCounts = h.BinCounts;

classNames = h.Categories;

查找包含少于十观察点的类。

idxLowCounts = classCounts < 10;

infrequentClasses = classNames(idxLowCounts)

infrequentClasses = 1×8 cell array

{'Freezing Fog'} {'Hurricane'} {'Lakeshore Flood'} {'Marine Dense Fog'} {'Marine Strong Wind'} {'Marine Tropical Depression'} {'Seiche'} {'Sneakerwave'}

从数据中删除这些不常见的类。使用removecats从分类数据中删除未使用的类别。

idxInfrequent = ismember(data.event\_type,infrequentClasses);

data(idxInfrequent,:) = [];

data.event\_type = removecats(data.event\_type);

现在, 数据被分类为合理大小的类。下一步是将其划分为训练、验证和测试的集合。将数据划分为训练分区和用于验证和测试的保留分区。指定维持百分比为30%。

cvp = cvpartition(data.event\_type,'Holdout',0.3);

dataTrain = data(training(cvp),:);

dataHeldOut = data(test(cvp),:);

再次对已保留的设置进行分区以获取验证集。指定维持百分比为50%。这将导致70% 种训练观察、15% 个验证观察和15% 个测试观察的分区。

cvp = cvpartition(dataHeldOut.event\_type,'HoldOut',0.5);

dataValidation = dataHeldOut(training(cvp),:);

dataTest = dataHeldOut(test(cvp),:);

从分区表中提取文本数据和标签。

textDataTrain = dataTrain.event\_narrative;

textDataValidation = dataValidation.event\_narrative;

textDataTest = dataTest.event\_narrative;

YTrain = dataTrain.event\_type;

YValidation = dataValidation.event\_type;

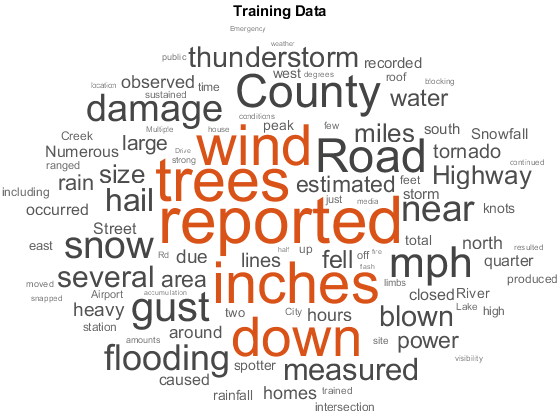
YTest = dataTest.event\_type;

若要检查是否已正确导入数据, 请使用 word 云可视化定型文本数据。

figure

wordcloud(textDataTrain);

title("Training Data")



### 预处理文本数据

预处理培训数据。将文本转换为小写, 标记化它, 然后擦除标点符号。不要词干或删除单词, 因为这些步骤会导致更糟糕的单词嵌入适合。

textDataTrain = lower(textDataTrain);

documentsTrain = tokenizedDocument(textDataTrain);

documentsTrain = erasePunctuation(documentsTrain);

textDataValidation = lower(textDataValidation);

documentsValidation = tokenizedDocument(textDataValidation);

documentsValidation = erasePunctuation(documentsValidation);

查看前几个经过预处理的培训文档。

documentsTrain(1:5)

ans =

5×1 tokenizedDocument:

(1,1) 7 tokens: large tree down between plantersville and nettleton

(2,1) 37 tokens: one to two feet of deep standing water developed on a stre…

(3,1) 13 tokens: nws columbia relayed a report of trees blown down along to…

(4,1) 13 tokens: media reported two trees blown down along i40 in the old f…

(5,1) 14 tokens: a few tree limbs greater than 6 inches down on hwy 18 in r…

### 将文档转换为序列

要将文档输入到 LSTM 网络中, 请使用 word 编码将文档转换为数字索引的序列。

若要创建单词编码, 请使用wordEncoding函数。

enc = wordEncoding(documentsTrain);

下一个转换步骤是对文档进行填充和截断, 以便它们的长度都相同。trainingOptions函数提供了自动填充和截断输入序列的选项。但是, 这些选项不太适合于单词向量的序列。而是手动填充和截断序列。如果你左垫和截断单词向量的序列, 那么训练可能会改进。

要对文档进行填充和截断, 请首先选择目标长度, 然后截断比它长的文档和比它短的左垫文档。为获得最佳效果, 目标长度应短于不丢弃大量数据。若要查找合适的目标长度, 请查看训练文档长度的直方图。

documentLengths = doclength(documentsTrain);

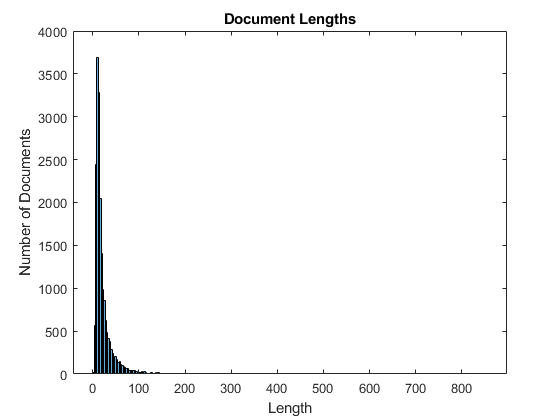
figure

histogram(documentLengths)

title("Document Lengths")

xlabel("Length")

ylabel("Number of Documents")



大多数培训文档的标记少于75个。将其用作截断和填充的目标长度。

使用doc2sequence将文档转换为数字索引的序列。要截断或向左填充序列的长度为 75, 请将'Length'选项设置为75。

XTrain = doc2sequence(enc,documentsTrain,'Length',75);

XTrain(1:5)

ans = 5×1 cell array

{1×75 double}

{1×75 double}

{1×75 double}

{1×75 double}

{1×75 double}

使用相同的选项将验证文档转换为序列。

XValidation = doc2sequence(enc,documentsValidation,'Length',75);

### 创建和培训 LSTM 网络

定义 LSTM 网络体系结构。要将序列数据输入到网络中, 请包括序列输入图层并将输入大小设置为1。接下来, 包括一个包含维度100的单词嵌入层, 以及与单词编码相同的单词数。接下来, 包括一个 LSTM 层, 并将隐藏单元的数量设置为180。要使用 LSTM 图层进行序列到标签分类问题, 请将输出模式设置为'last'。最后, 添加与类数、softmax 层和分类层大小相同的完全连接图层。

inputSize = 1;

embeddingDimension = 100;

numHiddenUnits = enc.NumWords;

hiddenSize = 180;

numClasses = numel(categories(YTrain));

layers = [ ...

sequenceInputLayer(inputSize)

wordEmbeddingLayer(embeddingDimension,numHiddenUnits)

lstmLayer(hiddenSize,'OutputMode','last')

fullyConnectedLayer(numClasses)

softmaxLayer

classificationLayer]

layers =

6x1 Layer array with layers:

1 '' Sequence Input Sequence input with 1 dimensions

2 '' Word Embedding Layer Word embedding layer with 100 dimensions and 16954 unique words

3 '' LSTM LSTM with 180 hidden units

4 '' Fully Connected 39 fully connected layer

5 '' Softmax softmax

6 '' Classification Output crossentropyex

指定培训选项。将解算器设置为'adam', 训练10纪元, 并将渐变阈值设置为1。将初始学习率设置为0.01。要监控培训进度, 请将'Plots'选项设置为'training-progress'。使用'ValidationData'选项指定验证数据。要抑制详细输出, 请将'Verbose'设置为false.

默认情况下, trainNetwork使用 gpu (如果可用) (需要并行计算 Toolbox™和具有计算能力3.0 或更高的 CUDA®启用 gpu)。否则, 它将使用 CPU。若要手动指定执行环境, 请使用trainingOptions的'ExecutionEnvironment'名称-值对参数。在 CPU 上进行的培训比在 GPU 上进行培训要长得多。

options = trainingOptions('adam', ...

'MaxEpochs',10, ...

'GradientThreshold',1, ...

'InitialLearnRate',0.01, ...

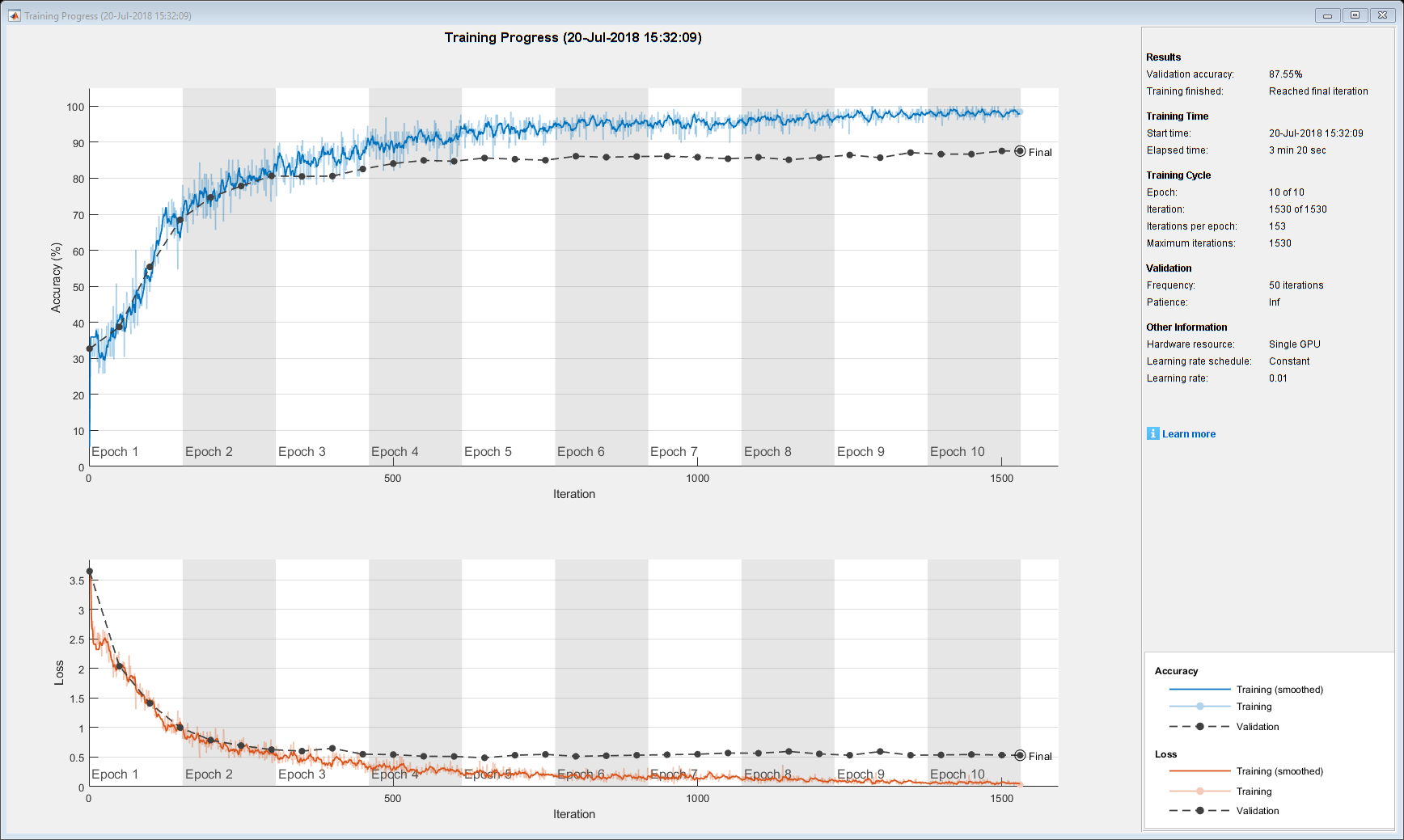
'ValidationData',{XValidation,YValidation}, ...

'Plots','training-progress', ...

'Verbose',false);

使用trainNetwork功能训练 LSTM 网络。

net = trainNetwork(XTrain,YTrain,layers,options);



### 测试 LSTM 网络

要测试 LSTM 网络, 首先要以与训练数据相同的方式准备测试数据。然后使用训练有素的 LSTM 网络net对预处理测试数据进行预测.

使用与培训文档相同的步骤预处理测试数据。

textDataTest = lower(textDataTest);

documentsTest = tokenizedDocument(textDataTest);

documentsTest = erasePunctuation(documentsTest);

使用doc2sequence将测试文档转换为序列, 方法与创建训练序列时的选项相同。

XTest = doc2sequence(enc,documentsTest,'Length',75);

XTest(1:5)

ans = 5×1 cell array

{1×75 double}

{1×75 double}

{1×75 double}

{1×75 double}

{1×75 double}

使用训练有素的 LSTM 网络对测试文档进行分类。

YPred = classify(net,XTest);

计算分类精度。准确性是网络预测正确的标签比例。

accuracy = sum(YPred == YTest)/numel(YPred)

accuracy = 0.8691

### 使用新数据预测

对三个新天气预报的事件类型进行分类。创建包含新天气报告的字符串数组。

reportsNew = [ ...

"Lots of water damage to computer equipment inside the office."

"A large tree is downed and blocking traffic outside Apple Hill."

"Damage to many car windshields in parking lot."];

使用与培训文档相同的步骤预处理文本数据。

reportsNew = lower(reportsNew);

documentsNew = tokenizedDocument(reportsNew);

documentsNew = erasePunctuation(documentsNew);

使用doc2sequence将文本数据转换为序列, 方法与创建训练序列时的选项相同。

XNew = doc2sequence(enc,documentsNew,'Length',75);

使用训练有素的 LSTM 网络对新序列进行分类。

[labelsNew,score] = classify(net,XNew);

使用预测的标签显示天气报告。

[reportsNew string(labelsNew)]

ans = 3×2 string array

"lots of water damage to computer equipment inside the office." "Flash Flood"

"a large tree is downed and blocking traffic outside apple hill." "Thunderstorm Wind"

"damage to many car windshields in parking lot." "Hail"

## 使用多词短语分析文本数据

此示例演示如何使用 n 克频率计数分析文本。

### n-克

n 克是一个元组的https://www.mathworks.com/help/examples/textanalytics/win64/AnalyzeTextDataUsingMultiWordPhrasesExample_eq01.png连续的单词。例如, 分词 (当https://www.mathworks.com/help/examples/textanalytics/win64/AnalyzeTextDataUsingMultiWordPhrasesExample_eq02.png) 是一对连续的词, 如 "大雨"。unigram (如果https://www.mathworks.com/help/examples/textanalytics/win64/AnalyzeTextDataUsingMultiWordPhrasesExample_eq03.png) 是一个单字。一个包的 n 克模型记录不同的 n 克在文档集合中出现的次数。

使用一包 n 克模型, 您可以保留有关原始文本数据中的单词排序的更多信息。例如, 一袋 n 克的模型更适合捕捉出现在文本中的短短语, 如 "暴雨" 和 "雷雨风"。

要创建一袋 n-克的模型, 请使用bagOfNgrams。您可以将bagOfNgrams对象输入到其他文本分析工具箱功能, 如wordcloud和fitlda.

### 加载和提取文本数据

若要重现此示例的结果, 请将rng设置为'default'.

rng('default')

加载示例数据。文件weatherReports.csv包含天气报告, 包括每个事件的文本说明和分类标签。删除具有空报表的行。

filename = "weatherReports.csv";

data = readtable(filename,'TextType','String');

idx = strlength(data.event\_narrative) == 0;

data(idx,:) = [];

从表中提取文本数据并查看前几个报表。

textData = data.event\_narrative;

textData(1:5)

ans = 5×1 string array

"Large tree down between Plantersville and Nettleton."

"One to two feet of deep standing water developed on a street on the Winthrop University campus after more than an inch of rain fell in less than an hour. One vehicle was stalled in the water."

"NWS Columbia relayed a report of trees blown down along Tom Hall St."

"Media reported two trees blown down along I-40 in the Old Fort area."

"A few tree limbs greater than 6 inches down on HWY 18 in Roseland."

### 准备用于分析的文本数据

创建先标记和预处理文本数据的函数, 以便用于分析。本示例末尾列出的函数preprocessWeatherNarratives执行以下步骤:

1. 使用lower的文本数据转换为小写.
2. 使用tokenizedDocument标记化文本.
3. 使用erasePunctuation擦除标点.
4. 使用removeStopWords删除停止字词 (如 "and"、"" 和 ") 的列表.
5. 使用removeShortWords删除2个或更少字符的单词.
6. 使用removeLongWords删除具有15个或更多字符的单词.
7. 用normalizeWords Lemmatize 单词.

使用示例预处理函数preprocessWeatherNarratives准备文本数据。

documents = preprocessWeatherNarratives(textData);

documents(1:5)

ans =

5×1 tokenizedDocument:

(1,1) 5 tokens: large tree down plantersville nettleton

(2,1) 18 tokens: two foot deep standing water develop street winthrop unive…

(3,1) 9 tokens: nws columbia relayed report tree blow down tom hall

(4,1) 10 tokens: medium report two tree blow down i40 old fort area

(5,1) 8 tokens: few tree limb great inches down hwy roseland

### 创建二的文字云

首先使用bagOfNgrams创建一个二模型, 然后将该模型输入到wordcloud , 从而创造一个词云。.

要计算长度为 2 (二) 的 n 克, 请使用默认选项bagOfNgrams 。

bag = bagOfNgrams(documents)

bag =

bagOfNgrams with properties:

Counts: [28138×117043 double]

Vocabulary: [1×18409 string]

Ngrams: [117043×2 string]

NgramLengths: 2

NumNgrams: 117043

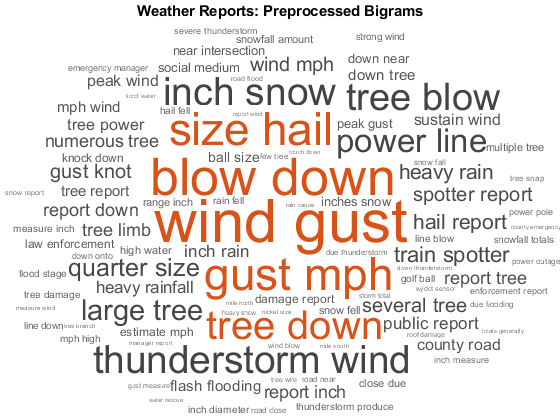
NumDocuments: 28138

使用单词云可视化包的 n 克模型。

figure

wordcloud(bag);

title("Weather Reports: Preprocessed Bigrams")



### 适合主题模型到袋的 n-克

潜在的不分配 (LDA) 模型是一个主题模型, 它发现文档集合中的基础主题, 并在主题中推断单词概率。

使用fitlda创建具有10个主题的 LDA 主题模型。该函数适用于 LDA 模型, 将 n 克作为单个单词进行处理。

mdl = fitlda(bag,10);

Initial topic assignments sampled in 0.741989 seconds.

=====================================================================================

| Iteration | Time per | Relative | Training | Topic | Topic |

| | iteration | change in | perplexity | concentration | concentration |

| | (seconds) | log(L) | | | iterations |

=====================================================================================

| 0 | 2.81 | | 2.043e+04 | 2.500 | 0 |

| 1 | 3.62 | 6.8345e-02 | 1.083e+04 | 2.500 | 0 |

| 2 | 3.54 | 1.9129e-03 | 1.064e+04 | 2.500 | 0 |

| 3 | 3.79 | 2.4671e-04 | 1.061e+04 | 2.500 | 0 |

| 4 | 3.81 | 8.5912e-05 | 1.060e+04 | 2.500 | 0 |

=====================================================================================

将前四个主题可视化为 word 云。

figure

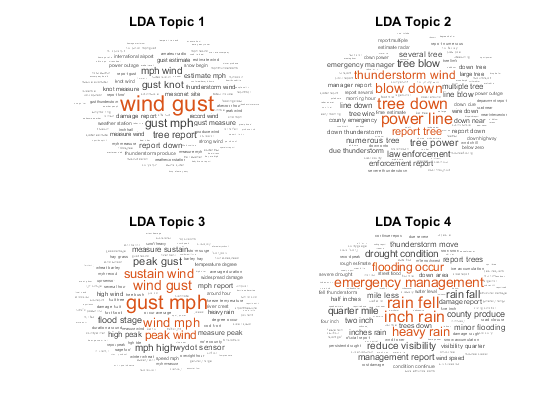
for i = 1:4

subplot(2,2,i)

wordcloud(mdl,i);

title("LDA Topic " + i)

end



"云" 一词突出显示了 LDA 主题中常见的二。该函数根据指定的 LDA 主题的概率绘制二的大小。

### 使用较长的短语分析文本

若要使用较长的短语分析文本, 请在bagOfNgrams中将'NGramLengths'选项指定为更大的值。

当使用较长的短语时, 在模型中保持停止单词可能很有用。例如, 要检测短语 "不快乐", 请在模型中保留 "是" 和 "不" 一词。

预处理文本。使用erasePunctuation擦除标点, 标记化使用tokenizedDocument.

cleanTextData = erasePunctuation(textData);

documents = tokenizedDocument(cleanTextData);

要计算长度为 3 (八卦) 的 n 克, 请使用bagOfNgrams并将'NGramLengths'指定为3。

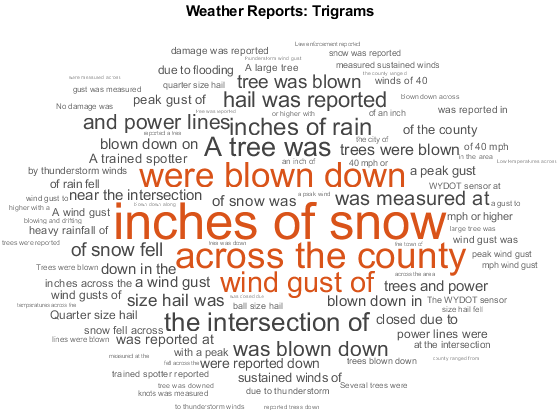
bag = bagOfNgrams(documents,'NGramLengths',3);

使用单词云可视化包的 n 克模型。八卦的字云更好地显示了各个单词的上下文。

figure

wordcloud(bag);

title("Weather Reports: Trigrams")



使用topkngrams查看前10八卦及其频率计数.

tbl = topkngrams(bag,10)

tbl=10×3 table

Ngram Count NgramLength

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

"inches" "of" "snow" 2075 3

"across" "the" "county" 1318 3

"were" "blown" "down" 1189 3

"wind" "gust" "of" 934 3

"A" "tree" "was" 860 3

"the" "intersection" "of" 812 3

"inches" "of" "rain" 739 3

"hail" "was" "reported" 648 3

"was" "blown" "down" 638 3

"and" "power" "lines" 631 3

### 预处理函数示例

函数preprocessWeatherNarratives按顺序执行以下步骤:

1. 使用lower的文本数据转换为小写.
2. 使用tokenizedDocument标记化文本.
3. 使用erasePunctuation擦除标点.
4. 使用removeStopWords删除停止字词 (如 "and"、"" 和 ") 的列表.
5. 使用removeShortWords删除2个或更少字符的单词.
6. 使用removeLongWords删除具有15个或更多字符的单词.
7. 用normalizeWords Lemmatize 单词.

function [documents] = preprocessWeatherNarratives(textData)

% Convert the text data to lowercase.

cleanTextData = lower(textData);

% Tokenize the text.

documents = tokenizedDocument(cleanTextData);

% 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 greater

% characters.

documents = removeShortWords(documents,2);

documents = removeLongWords(documents,15);

% Lemmatize the words.

documents = addPartOfSpeechDetails(documents);

documents = normalizeWords(documents,'Style','lemma');

end

## 训练情绪分类器

本示例演示如何使用带有正负情绪词和预先训练字嵌入的注释列表训练用于情绪分析的分类器。

预先训练字嵌入在该工作流中扮演几个角色。它将单词转换为数字向量, 并构成分类器的基础。然后, 您可以使用分类器来预测其他字词使用其矢量表示形式的情绪, 并使用这些分类来计算一段文本的情感。培训和使用情感分类器有四步:

* 加载预先训练字嵌入。
* 载入一个观点词汇表, 列出正负词。
* 用正负词的词向量训练情绪分类器。
* 计算文本中单词的平均情绪分数。

若要重现此示例中的结果, 请将rng设置为'default'.

rng('default')

### 加载预先训练字嵌入

Word 嵌入将词汇中的单词映射到数字向量。这些嵌入可以捕获单词的语义细节, 以便类似的词具有相似的向量。它们还通过向量算法对单词之间的关系进行建模。例如, 国王对女王的关系是男人对女人的描述由等式国王 - 男人 + 女人 = 皇后.

使用fastTextWordEmbedding函数加载预先训练字嵌入。此函数需要 fastText 英语160亿令牌字嵌入支持包的文本分析 Toolbox™模型。如果未安装此支持包, 则该函数将提供下载链接。

emb = fastTextWordEmbedding;

### 加载意见词典

从[https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html](https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon)的观点词汇 (也称为情感词汇) 中加载正负词。首先, 将.rar文件中的文件提取到名为 "opinion-lexicon-English" 的文件夹中, 然后导入文本。

使用本示例末尾列出的函数readLexicon加载数据。输出data是包含单词的变量Word的表,Label包含分类情感标签,Positive或Negative.

data = readLexicon;

查看标记为正数的前几个单词。

idx = data.Label == "Positive";

head(data(idx,:))

ans=8×2 table

Word Label

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

"a+" Positive

"abound" Positive

"abounds" Positive

"abundance" Positive

"abundant" Positive

"accessable" Positive

"accessible" Positive

"acclaim" Positive

查看标记为负数的前几个单词。

idx = data.Label == "Negative";

head(data(idx,:))

ans=8×2 table

Word Label

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

"2-faced" Negative

"2-faces" Negative

"abnormal" Negative

"abolish" Negative

"abominable" Negative

"abominably" Negative

"abominate" Negative

"abomination" Negative

### 准备培训数据

为了训练情绪分类器, 使用预先训练字嵌入emb局将单词转换为词向量。首先删除不出现在 word 中的单词嵌入emb局.

idx = ~isVocabularyWord(emb,data.Word);

data(idx,:) = [];

随机放置10% 个单词进行测试。

numWords = size(data,1);

cvp = cvpartition(numWords,'HoldOut',0.1);

dataTrain = data(training(cvp),:);

dataTest = data(test(cvp),:);

使用word2vec将训练数据中的单词转换为词向量.

wordsTrain = dataTrain.Word;

XTrain = word2vec(emb,wordsTrain);

YTrain = dataTrain.Label;

### 训练情绪分类器

训练支持向量机 (SVM) 分类器, 将词向量分类为正类和负类别。

mdl = fitcsvm(XTrain,YTrain);

### 测试分类器

使用word2vec将测试数据中的单词转换为 word 向量.

wordsTest = dataTest.Word;

XTest = word2vec(emb,wordsTest);

YTest = dataTest.Label;

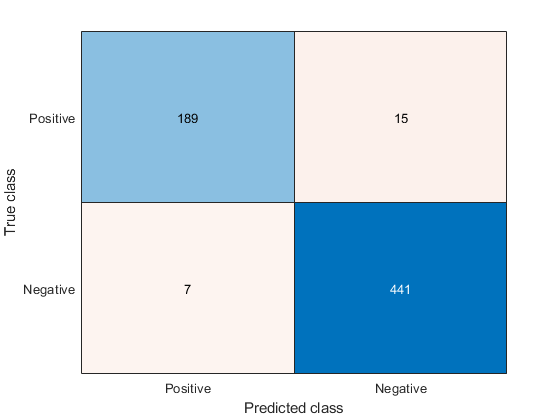
预测测试词向量的情感标签。

[YPred,scores] = predict(mdl,XTest);

在混淆矩阵中可视化分类精度。

figure

confusionchart(YTest,YPred);



在 word 云中可视化分类。用与预测分数对应的单词大小在 word 云中绘制带有正负情绪的单词。

figure

subplot(1,2,1)

idx = YPred == "Positive";

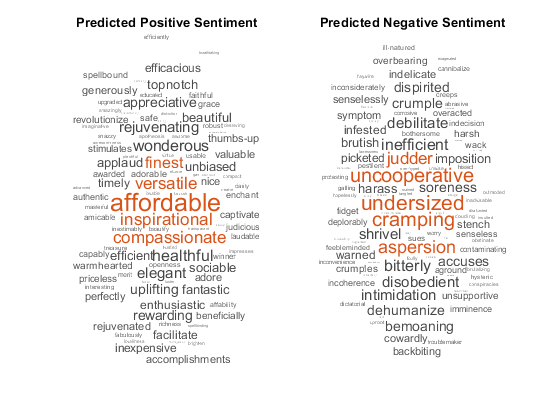
wordcloud(wordsTest(idx),scores(idx,1));

title("Predicted Positive Sentiment")

subplot(1,2,2)

wordcloud(wordsTest(~idx),scores(~idx,2));

title("Predicted Negative Sentiment")



### 计算文本集合的情绪

为了计算一段文本的情感, 例如复习, 预测文本中每个单词的情绪得分, 并采取平均的情绪评分。

加载 Airbnb 总结审查数据 (波士顿, 马萨诸塞州, 美国, 06 2017年10月) 从<http://insideairbnb.com/get-the-data.html>。将数据读取到表中, 并指定将文本数据作为字符串读取。

filename = "reviews.csv";

dataReviews = readtable(filename,'TextType','string');

从comments变量中提取文本数据并查看前几个评论。

textData = dataReviews.comments;

textData(1:10)

ans = 10×1 string array

"Pretty nice, quiet, cozy place to stay. Toiletries, snacks, coffee, WiFi, cable TV, iron was all included. One of the best things for me is how quiet it was even in the daytime. Coded door locks so no need for keys, my belongings were always safe and Andre and his wife are really good host. I stayed 7 days and never had a problem. I'll stay again if and when I had the chance."

"The host was extremely welcoming and obliging. The neighborhood is quiet and charming, perfect for a quiet visit. Short walk to MBTA transportation."

"Nice and easy stay - with good accommodations especially the cable TV "

"The host has been very accommodating and helpful. The description in the ad is accurate. The room is very clean and the neighborhood is quiet."

"It's a great quiet stay."

"Couldn't have been happier. The apartment was well renovated, very clean and convenient to great spots. The kitchen was stocked with all the basics and a huge grocery store was around the corner so we were able to easily cook at the house. Estee also provided some great local recommendations. Wine, snacks, coffee and games were great extras. Uber ride to downtown was $8. Would most definitely stay here again."

"The apartment is very nice- as described and very convenient. The real superstar of the listing though is the host; Estee was phenomenal. She was very responsive and even let us know when she might not be able to be reached for a short duration of time. She provided great recommendations and tips for getting around. We had a MINOR issue, which she went out of her way to resolve very quickly. ↵↵Both bedrooms are a good size, and one has a lovely vanity. Everything is brand new - bathroom and kitchen. Estee had the kitchen stocked with staples (salt, pepper, olive oil, ketchup) and treats too! There are so many details throughout the place where she goes above and beyond. Parking on the street was easy. We hardly needed to move the car though because there was so much within walking distance. The description of a 10 minute walk to the T is accurate. ↵↵100% would stay here again. Thank you for a wonderful stay, Estee!"

"This is a brand new gorgeous place, very clean, bright and welcoming. Estee especially knows how to make guests comfortable, there were many thoughtful touches and she recommended a delicious Indian restaurant. There is a supermarket within 5 minutes walking distance and we used Uber to get around - downtown Boston took less than 15 minutes. Best place I have stayed in so far. Thank you Estee!"

"Estee and Josh are great hosts. Very welcoming. Made us feel like we were staying with long time friends. Apartment very centrally located. Off street parking surprisingly easy (for Boston). Loads of restaurants within walking distance"

"Estee was super sweet and so very accommodating! The apartment was nicely renovated and the kitchen had all our basic needs + treats as well! My family and I stayed here because of a college graduation and because street parking in front of her place was easy and everything was within walking distance, it made our stay a lot easier! Would definitely stay here again! "

创建先标记和预处理文本数据的函数, 以便用于分析。在示例末尾列出的函数preprocessReviews按顺序执行以下步骤:

1. 使用lower的文本数据转换为小写.
2. 使用tokenizedDocument标记化文本.
3. 使用erasePunctuation擦除标点.
4. 使用removeStopWords删除停止字词 (如 "和"、"和").

使用预处理函数preprocessReviews准备文本数据。此步骤可能需要几分钟时间才能运行。

documents = preprocessReviews(textData);

删除文档中未出现的单词嵌入emb局.

idx = ~isVocabularyWord(emb,documents.Vocabulary);

documents = removeWords(documents,idx);

为了可视化情绪分类器对评论的概括程度, 对评论中出现的字词进行分类, 而不是在训练数据中对其进行可视化, 并在 word 云中对其进行形象化。使用 word 云手动检查分类器的行为是否按预期的方式。

words = documents.Vocabulary;

words(ismember(words,wordsTrain)) = [];

vec = word2vec(emb,words);

[YPred,scores] = predict(mdl,vec);

figure

subplot(1,2,1)

idx = YPred == "Positive";

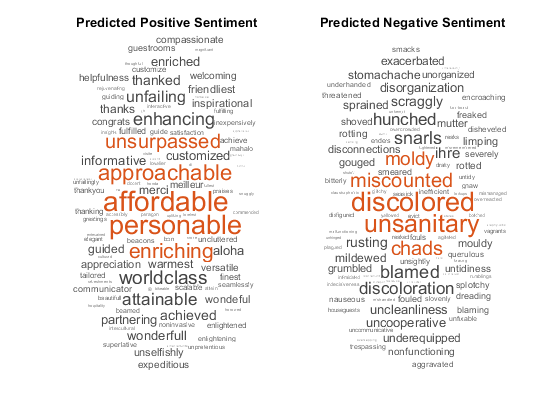
wordcloud(words(idx),scores(idx,1));

title("Predicted Positive Sentiment")

subplot(1,2,2)

wordcloud(words(~idx),scores(~idx,2));

title("Predicted Negative Sentiment")



要计算给定文本的情绪, 请计算文本中每个单词的情感分数, 并计算平均情绪得分。

对于选择的文档, 计算平均情绪评分。对于每个文档, 将单词转换为词向量, 预测词向量上的情感分数, 使用分数-后转换函数转换分数, 然后计算平均情绪分数。

idx = [7 34 331 1788 1820 1831 2185 21892 63734 76832 113276 120210];

for i = 1:numel(idx)

words = string(documents(idx(i)));

vec = word2vec(emb,words);

[~,scores] = predict(mdl,vec);

sentimentScore(i) = mean(scores(:,1));

end

使用文本数据查看预测的情绪得分。分数大于0对应于积极情绪, 分数小于0对应于负面情绪, 分数接近0对应于中性情绪。

[sentimentScore' textData(idx)]

ans = 12×2 string array

"0.85721" "The apartment is very nice- as described and very convenient. The real superstar of the listing though is the host; Estee was phenomenal. She was very responsive and even let us know when she might not be able to be reached for a short duration of time. She provided great recommendations and tips for getting around. We had a MINOR issue, which she went out of her way to resolve very quickly. ↵↵Both bedrooms are a good size, and one has a lovely vanity. Everything is brand new - bathroom and kitchen. Estee had the kitchen stocked with staples (salt, pepper, olive oil, ketchup) and treats too! There are so many details throughout the place where she goes above and beyond. Parking on the street was easy. We hardly needed to move the car though because there was so much within walking distance. The description of a 10 minute walk to the T is accurate. ↵↵100% would stay here again. Thank you for a wonderful stay, Estee!"

"2.0453" "Estee was the perfect Airbnb host. The apartment was comfortable, spacious, and convenient, and Estee went to great lengths to make sure that we felt at home. She also provided great tips for us about the area. Would definitely love to stay here again."

"-0.37918" "The apartment is not apropriate for 5 people. Is too little and We were no comfortable. The bathroom was no clean. There was a door in the kitchen Broken. Is Too noisy. The elevator is Too small just for 2 people. "

"0.94799" "Truly a quaint place in Beacon Hill. Comfortable walking distance from MGH, Boston Common, and Suffolk University. The studio type place is great for a couple's weekend. The wifi was excellent as was the tv and comfort of the bed. ↵The limitations and recommendations for improvement include:↵1- improving in cleanliness as the floor was dirty enough that you couldn't walk around without shoes↵2- would recommend bringing your own basic toiletries as there was no hand soap in the bathroom. (We were too busy to contact jj but he is quick to respond to other matters so maybe he would have made arrangements to provide you with it.)↵3- storage space is limited so a prolonged stay would be challenging↵Overall, this a very functional stay."

"-0.077053" "the neibourhood is perfect!!!!!. as it is very close to Bowdoin T STATION and Park T station, walking distance from everywhere we wanted to go, quincy market, downtown, chinatown, newsbury street and every thing. the appartment IT IS NOT on Temple street rear... it is on Coolidge st, facing a quite silent and lonely and big parking lot. (:/) it was ok...though. coming and going was easy. JJ was really quick responsive when internet and CAble TV suddenly stopped but he was very helpful trying to solve it. that was very good. ... Other issues: By the house roules and the descriptions of previews guests I supposed the appartment was inmaculated and the cleaning was really fond... BUT IT WASNÂ´T. we found previous litter in the trash bins... kitchen and bathroom... the brown chocolate cuchions didnÂ´t smell as if they were clean. There were uncovered sheets, and blanquets and who knows what else under the bed, that I tried not to sweep the floor in its direction in case I made them dirty with the dust and gravel that was already inside when we got into the apparment. I went to the closet looking for the broom and shovel and I found them... the broom plenty of dirt and lint and entangled long hairs and stuff, and the shovel broken... very discusting. It is a pitty such an amazing location dealing with all these ackward details that are not ok at all. I think everything I mentioned can be solved easily in a very simple and cheap and loving form so the place becomes the perfect spot to spent your vacations in Boston."

"0.17846" "Although we didn't meet JJ, we felt he was very quick to respond. Checking in and out was a breeze. The location is convenient and walkable to everything we wanted to see. Studio was small but certainly comfortable for the 2 of us. ↵We didn't see any info regarding wifi (nor did we ask) so we didn't use it. The bathroom sink drain looked pretty dirty, no hand soap or wash cloths. It's a basement so there is not much natural light which was fine but even the lights in the space still made it feel very dark. We also found no place to hang our stuff or stash our clothes. A lot of the drawers had clothes in them already. ↵All in all, we liked the place. Would recommend it if a few changeable things were addressed. "

"-0.31603" "In the apartment it was very dirty .↵we walked in and there instincts to rotten melon .↵the sink was full of dirty dishes.↵the microwave and mini oven were dirty you could make it nothing to eat.↵on the herd was a coffeepot with moldy coffee.↵on the first day of our "vacation" we first had to make everything clean that we can feel comfortable.↵carissa wanted the city to show us what to do something .↵But in the week she was not at home , and when they came home she was for days in her room she only came out to make himself something to eat and dishes piled up again."

"-4.0895" "Blackmail!"

"1.658" "Outstanding stay. The apartment is world-class - very, very nice. Tremendous views of the harbor from a very cool apartment in a very cool, brand new retro building. I had not spent time in the South Boston waterfront neighborhood previously and loved it - great cafes, restaurants, pubs, renovated lofts. A terrific area.↵↵In addition, John was an ideal host. Incredibly responsive and helpful. Provided excellent recommendations in terms of spots to visit in the neighborhood as well as very clear directions relating to the logistics of checking in, wifi access, heating/cooling, etc.↵↵Finally, John is an engaging and interesting person who is an absolute pleasure to spend time with."

"1.7102" "I had an amazing stay at Carney's. The hosts are friendly and very meticulous. They made sure everything was proper from the kitchen needs to the bedroom needs. Also, they made us feel like we are at home. My parents had come for my graduation and they were pleased that I did not book a hotel and instead chose to stay here. I would recommend everyone to book a room if ever they plan to come to Boston and enjoy an enriching confortable experience. "

"0.67654" "My husband and I came to Boston for our 1 year anniversary. We're so glad we found Elisabeth's place! Immediately when we got dropped off by our cab, Elisabeth came outside and walked us to the house. She's incredibly nice, personable, and her place was beautiful and very clean. We felt very comfortable staying with her and she was nice enough to give us a few recommendations around town. Her place is a short walk to the Red Line T station and the neighborhood where she lives has everything you need close by. Gas station and convenience store was literally down the street and lots of smaller mom and pop restaurants. We hated to leave so early but would definitely love to come back! "

"-0.21651" "My fiancÃ© and I had just gotten engaged and wanted to stay somewhere a bit more upscale for our last night in Boston. We looked and found this "penthouse" and from arrival were let down. While the host was pleasant enough, she was hard to contact, the address was wrong, and she even had to have a neighbor show us around the place. Which would not have been weird if he wasn't doing laundry during our stay. We were promised the entire condo but the host stopped by as well, not that we minded that part, but it added to the weirdness. We were not able to use the refrigerator to store leftovers due to the HORRIBLE smell coming from it. It was so bad we turned the air off and opened the little balcony door. The condo looked too loved in to justify paying what we did. Also very confused about the $50 cleaning fee that was obviously not used before our stay, so a bit unhappy that we overpaid for a dirty place. "

### 情感词汇阅读功能

此函数从情感词汇表中读取正负词并返回表格。该表包含变量Word和Label, 其中Label包含与每个单词的情绪对应的Positive和Negative的分类值。

function data = readLexicon

% Read positive words

fidPositive = fopen(fullfile('opinion-lexicon-English','positive-words.txt'));

C = textscan(fidPositive,'%s','CommentStyle',';');

wordsPositive = string(C{1});

% Read negative words

fidNegative = fopen(fullfile('opinion-lexicon-English','negative-words.txt'));

C = textscan(fidNegative,'%s','CommentStyle',';');

wordsNegative = string(C{1});

fclose all;

% Create table of labeled words

words = [wordsPositive;wordsNegative];

labels = categorical(nan(numel(words),1));

labels(1:numel(wordsPositive)) = "Positive";

labels(numel(wordsPositive)+1:end) = "Negative";

data = table(words,labels,'VariableNames',{'Word','Label'});

end

### 预处理功能

函数preprocessReviews执行以下步骤:

1. 使用lower的文本数据转换为小写.
2. 使用tokenizedDocument标记化文本.
3. 使用erasePunctuation擦除标点.
4. 使用removeStopWords删除停止字词 (如 "和"、"和").

function [documents] = preprocessReviews(textData)

% Convert the text data to lowercase.

cleanTextData = lower(textData);

% Tokenize the text.

documents = tokenizedDocument(cleanTextData);

% Erase punctuation.

documents = erasePunctuation(documents);

% Remove a list of stop words.

documents = removeStopWords(documents);

end

## 使用自定义小型批量数据存储对内存不足的文本数据进行分类

此示例使用:

* [深度学习工具箱](https://www.mathworks.com/help/deeplearning/index.html)
* [文本分析工具箱](https://www.mathworks.com/help/textanalytics/index.html)

此示例演示如何使用自定义小型批量数据存储将内存不足的文本数据与深度学习网络分类。

小型批量数据存储是一个数据存储的实现, 它支持批量读取数据。您可以将小型批量数据存储用作深度学习应用程序的培训、验证、测试和预测数据集的来源。在读取批数据时, 使用小型批处理数据存储来读取内存不足或执行特定的预处理操作。

在训练网络时, 软件通过填充、截断或分割输入数据来创建具有相同长度的小批量序列。trainingOptions函数提供了用于填充和截断输入序列的选项, 但是, 这些选项并不适合于单词向量的序列。此外, 此函数不支持自定义数据存储中的填充信息。相反, 您必须手动填充和截断序列。如果你左垫和截断单词向量的序列, 那么训练可能会改进。

[使用深度学习对文本数据进行分类](https://www.mathworks.com/help/textanalytics/ug/classify-text-data-using-deep-learning.html)会手动截断所有文档并将其填充到相同的长度。此过程将大量填充添加到非常短的文档中, 并从很长的文档中丢弃大量数据。

或者, 为了防止添加过多的填充或丢弃过多的数据, 请创建一个自定义的小型批处理数据存储, 将小批量输入到网络中。自定义微型批处理数据存储textDatastore.m将小批量的文档转换为序列或 word 索引, 并将每个小批量的文件左填充到小型批处理中最长文档的长度。对于已排序的数据, 此数据存储可以帮助减少添加到数据中的填充量, 因为文档未填充到固定长度。同样, 数据存储不会丢弃文档中的任何数据。

本示例使用自定义小型批处理数据存储[textDatastore.m](matlab:edit(fullfile(matlabroot,'examples','nnet','main','textDatastore.m')))。通过自定义函数, 可以将此数据存储调整到您的数据库。有关演示如何创建自己的自定义小型批量数据存储的示例, 请参阅[开发自定义小型批量数据存储](https://www.mathworks.com/help/deeplearning/ug/develop-custom-mini-batch-datastore.html)(深度学习工具箱)。

### 加载预先训练字嵌入

数据存储textDatastore需要嵌入一个字来将文档转换为矢量序列。使用fastTextWordEmbedding加载预先训练字嵌入。此函数需要 fastText 英语160亿令牌字嵌入支持包的文本分析 Toolbox™模型。如果未安装此支持包, 则该函数将提供下载链接。

emb = fastTextWordEmbedding;

### 创建文档的小批量数据存储

创建包含用于培训的数据的存储。自定义微型批处理数据存储textDatastore从 CSV 文件读取预测因子和标签。对于预测变量, 数据存储将文档转换为 word 索引和响应的序列, 数据存储为每个文档返回分类标签。

若要创建数据存储, 请首先将自定义小型批处理数据存储[textDatastore.m](matlab:edit(fullfile(matlabroot,'examples','nnet','main','textDatastore.m')))保存到路径中。有关创建自定义小型批处理数据存储的详细信息, 请参阅[开发自定义小型批处理存储](https://www.mathworks.com/help/deeplearning/ug/develop-custom-mini-batch-datastore.html)(深度学习工具箱)。

对于培训数据, 请指定 csv 文件"weatherReportsTrain.csv" , 并且文本和标签分别位于"event\_narrative"和"event\_type"列中。

filenameTrain = "weatherReportsTrain.csv";

textName = "event\_narrative";

labelName = "event\_type";

dsTrain = textDatastore(filenameTrain,textName,labelName,emb)

dsTrain =

textDatastore with properties:

Datastore: [1×1 matlab.io.datastore.TabularTextDatastore]

TextName: "event\_narrative"

LabelName: "event\_type"

Classes: [1×39 string]

NumClasses: 39

Embedding: [1×1 wordEmbedding]

MiniBatchSize: 128

NumObservations: 19683

使用相同的步骤从 csv 文件"weatherReportsValidation.csv"创建包含验证数据的存储。

filenameValidation = "weatherReportsValidation.csv";

dsValidation = textDatastore(filenameValidation,textName,labelName,emb)

dsValidation =

textDatastore with properties:

Datastore: [1×1 matlab.io.datastore.TabularTextDatastore]

TextName: "event\_narrative"

LabelName: "event\_type"

Classes: [1×39 string]

NumClasses: 39

Embedding: [1×1 wordEmbedding]

MiniBatchSize: 128

NumObservations: 4218

### 创建和培训 LSTM 网络

定义 LSTM 网络体系结构。要将序列数据输入到网络中, 请包括序列输入层, 并将输入大小设置为嵌入维度。接下来, 包括一个 LSTM 层, 并将隐藏大小指定为180。要使用 LSTM 图层进行序列到标签分类问题, 请将输出模式设置为'last'。最后, 添加一个完全连接的层, 输出大小等于类数、softmax 层和分类层。

inputSize = dsTrain.Embedding.Dimension;

hiddenSize = 180;

numClasses = dsTrain.NumClasses;

layers = [ ...

sequenceInputLayer(inputSize)

lstmLayer(hiddenSize,'OutputMode','last')

fullyConnectedLayer(numClasses)

softmaxLayer

classificationLayer];

指定培训选项。指定求解器为'adam' , 渐变阈值为1。将初始学习率设置为0.01。数据存储textDatastore.m不支持洗牌, 因此将'Shuffle'" 设置为'never' (有关演示如何实现具有洗牌支持的数据存储的示例, 请参阅[开发自定义小型批量数据存储](https://www.mathworks.com/help/deeplearning/ug/develop-custom-mini-batch-datastore.html)(深度学习工具箱))。使用'ValidationData'选项指定验证数据。要监控培训进度, 请将'Plots'选项设置为'training-progress'。要抑制详细输出, 请将'Verbose'设置为false.

默认情况下, trainNetwork使用 gpu (如果可用) (需要并行计算 Toolbox™和具有计算能力3.0 或更高的 CUDA®启用 gpu)。否则, 它将使用 CPU。若要手动指定执行环境, 请使用trainingOptions的'ExecutionEnvironment'名称-值对参数。在 CPU 上进行的培训比在 GPU 上进行培训要长得多。

options = trainingOptions('adam', ...

'GradientThreshold',1, ...

'InitialLearnRate',0.01, ...

'Shuffle','never', ...

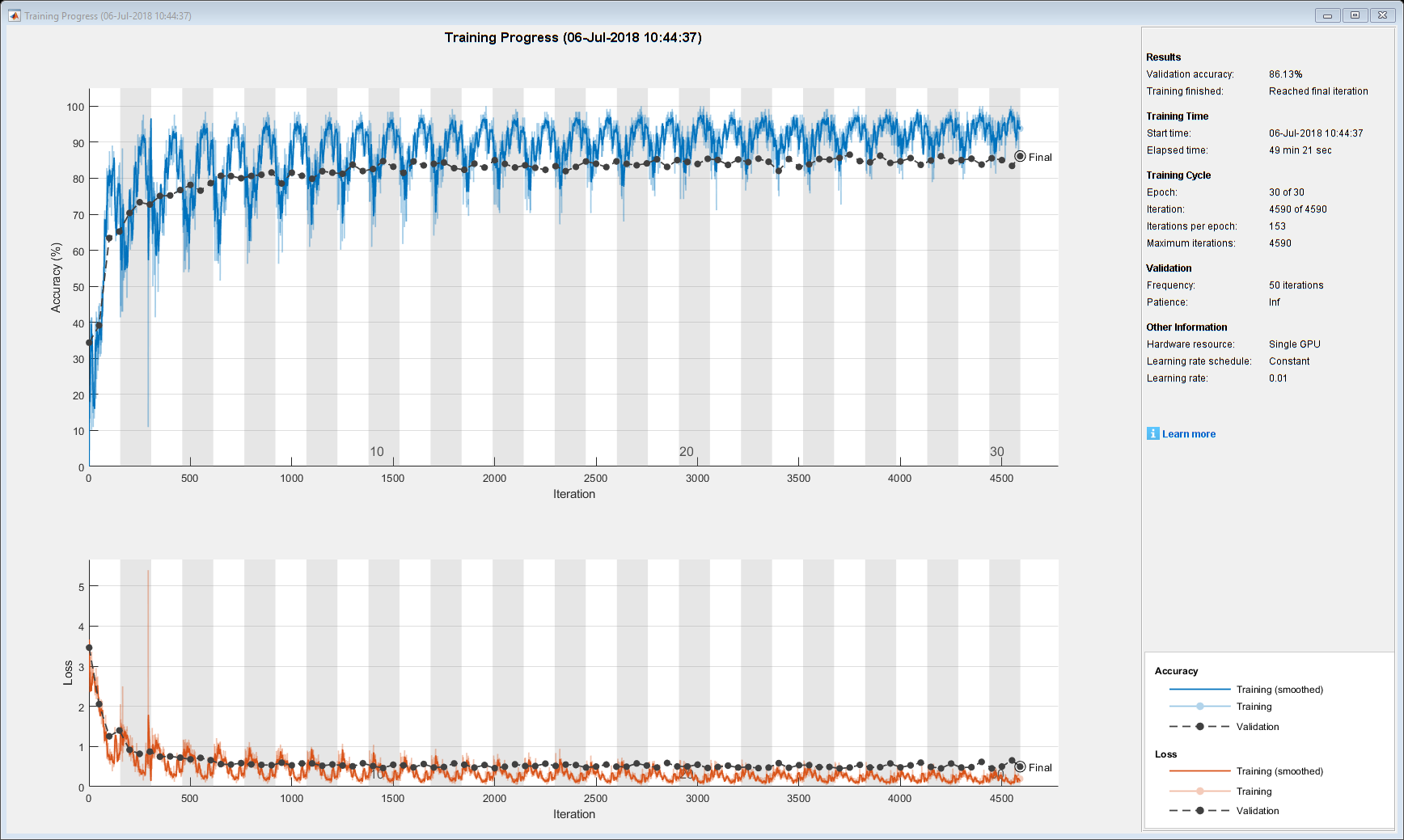
'Plots','training-progress', ...

'Verbose',false, ...

'ValidationData',dsValidation);

使用trainNetwork功能训练 LSTM 网络。

net = trainNetwork(dsTrain,layers,options);



### 测试 LSTM 网络

创建包含文档和标签的数据存储

filenameTest = "weatherReportsTest.csv";

dsTest = textDatastore(filenameTest,textName,labelName,emb)

dsTest =

textDatastore with properties:

Datastore: [1×1 matlab.io.datastore.TabularTextDatastore]

TextName: "event\_narrative"

LabelName: "event\_type"

Classes: [1×39 string]

NumClasses: 39

Embedding: [1×1 wordEmbedding]

MiniBatchSize: 128

NumObservations: 4217

使用训练有素的 LSTM 网络对测试文档进行分类。

YPred = classify(net,dsTest);

计算分类精度。准确性是网络预测正确的标签比例。

YTest = readLabels(dsTest);

accuracy = sum(YPred == YTest)/numel(YPred)

accuracy = 0.8328

## 选择 LDA 模型的主题数

此示例演示如何为潜在的的不分配 (LDA) 模型确定合适数量的主题。

要确定合适数量的主题, 您可以比较 LDA 模型适合于不同数量主题的拟合优度。您可以通过计算被保留的文档集的困惑来评估 LDA 模型的拟合性。困惑表明模型描述一组文档的程度。更低的困惑意味着更好的适应。

若要重现此示例的结果, 请将rng设置为'default'.

rng('default')

### 提取和预处理文本数据

加载示例数据。文件weatherReports.csv包含天气报告, 包括每个事件的文本说明和分类标签。从字段中提取文本数据event\_narrative.

filename = "weatherReports.csv";

data = readtable(filename,'TextType','string');

textData = data.event\_narrative;

使用本示例末尾列出的函数preprocessWeatherNarratives标记化和预处理文本数据。

documents = preprocessWeatherNarratives(textData);

documents(1:5)

ans =

5×1 tokenizedDocument:

(1,1) 5 tokens: large tree down plantersville nettleton

(2,1) 18 tokens: two foot deep standing water develop street winthrop unive…

(3,1) 9 tokens: nws columbia relayed report tree blow down tom hall

(4,1) 10 tokens: medium report two tree blow down i40 old fort area

(5,1) 0 tokens:

随机预留10% 份文件进行验证。

numDocuments = numel(documents);

cvp = cvpartition(numDocuments,'HoldOut',0.1);

documentsTrain = documents(cvp.training);

documentsValidation = documents(cvp.test);

从培训文档中创建一个单词袋模型。删除不超过两次出现的单词总数。删除任何不含文字的文档。

bag = bagOfWords(documentsTrain);

bag = removeInfrequentWords(bag,2);

bag = removeEmptyDocuments(bag);

### 选择主题数量

其目标是选择一些主题, 以最小化困惑是最低的与其他数量的主题。这不是唯一的考虑因素: 与大量主题匹配的模型可能需要更长的时间来收敛。要查看折衷的效果, 请计算拟合优度和拟合时间。如果主题的最佳数量很高, 则可能需要选择较低的值以加快拟合过程。

适用于某些 LDA 模型, 以获得主题数量的一系列值。将每个模型的拟合时间和困惑与测试文档的保留集进行比较。困惑是logp函数的第二个输出。若要获取第二个输出而不将第一个输出分配给任何内容, 请使用~符号。拟合时间是上次迭代的TimeSinceStart值。此值位于 LDA 模型的FitInfo属性的History结构中。

为了更快地适应, 请指定'Solver'为'savb'。要抑制详细输出, 请将'Verbose'设置为0。这可能需要几分钟时间才能运行。

numTopicsRange = [5 10 15 20 40];

for i = 1:numel(numTopicsRange)

numTopics = numTopicsRange(i);

mdl = fitlda(bag,numTopics, ...

'Solver','savb', ...

'Verbose',0);

[~,validationPerplexity(i)] = logp(mdl,documentsValidation);

timeElapsed(i) = mdl.FitInfo.History.TimeSinceStart(end);

end

显示情节中每个主题的困惑和经过时间。绘制左轴上的困惑和右轴上经过的时间。

figure

yyaxis left

plot(numTopicsRange,validationPerplexity,'+-')

ylabel("Validation Perplexity")

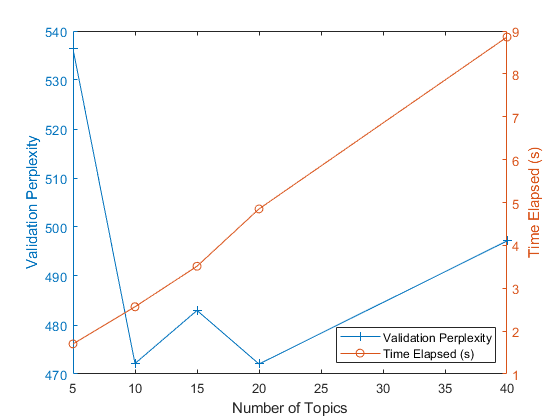
yyaxis right

plot(numTopicsRange,timeElapsed,'o-')

ylabel("Time Elapsed (s)")

legend(["Validation Perplexity" "Time Elapsed (s)"],'Location','southeast')

xlabel("Number of Topics")



剧情表明, 用10–20主题拟合模型可能是一个不错的选择。与不同主题数量的模型相比, 这种困惑是很低的。使用此解算器, 这许多主题的经过时间也是合理的。使用不同的求解器, 您可能会发现增加主题的数量可能会导致更适合, 但拟合模型需要更长的时间才能收敛。

### 预处理函数示例

函数preprocessWeatherNarratives, 按顺序执行以下步骤:

1. 使用lower的文本数据转换为小写.
2. 使用tokenizedDocument标记化文本.
3. 使用erasePunctuation擦除标点.
4. 使用removeStopWords删除停止字词 (如 "and"、"" 和 ") 的列表.
5. 使用removeShortWords删除2个或更少字符的单词.
6. 使用removeLongWords删除具有15个或更多字符的单词.
7. 用normalizeWords Lemmatize 单词.

function [documents] = preprocessWeatherNarratives(textData)

% Convert the text data to lowercase.

cleanTextData = lower(textData);

% Tokenize the text.

documents = tokenizedDocument(cleanTextData);

% 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 greater

% characters.

documents = removeShortWords(documents,2);

documents = removeLongWords(documents,15);

% Lemmatize the words.

documents = addPartOfSpeechDetails(documents);

documents = normalizeWords(documents,'Style','lemma');

end

# 显示和演示

## 使用 Word 云可视化文本数据

此示例演示如何使用 word 云可视化文本数据。

文本分析工具箱扩展了wordcloud (MATLAB) 函数的功能。它添加了对从字符串数组中直接创建 word 云的支持, 以及从单词袋模型和 LDA 主题创建 word 云。

加载示例数据。文件weatherReports.csv包含天气报告, 包括每个事件的文本说明和分类标签。

filename = "weatherReports.csv";

T = readtable(filename,'TextType','string');

从event\_narrative列中提取文本数据。

textData = T.event\_narrative;

textData(1:10)

ans = 10x1 string array

"Large tree down between Plantersville and Nettleton."

"One to two feet of deep standing water developed on a street on the Winthrop University campus after more than an inch of rain fell in less than an hour. One vehicle was stalled in the water."

"NWS Columbia relayed a report of trees blown down along Tom Hall St."

"Media reported two trees blown down along I-40 in the Old Fort area."

""

"A few tree limbs greater than 6 inches down on HWY 18 in Roseland."

"Awning blown off a building on Lamar Avenue. Multiple trees down near the intersection of Winchester and Perkins."

"Quarter size hail near Rosemark."

"Tin roof ripped off house on Old Memphis Road near Billings Drive. Several large trees down in the area."

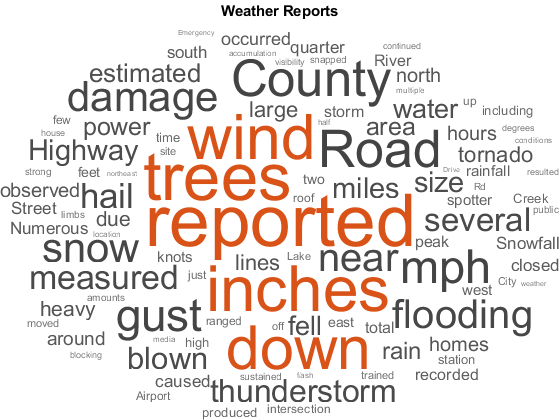
"Powerlines down at Walnut Grove and Cherry Lane roads."

从所有天气报告中创建一个单词云。

figure

wordcloud(textData);

title("Weather Reports")



将报告中的单词与"Hail"和"Thunderstorm Wind"标签进行比较。为每个标签创建报告的文字云。分别为每个单词云指定蓝色和洋红的单词颜色。

figure

labels = T.event\_type;

subplot(1,2,1)

idx = labels == "Hail";

wordcloud(textData(idx),'Color','blue');

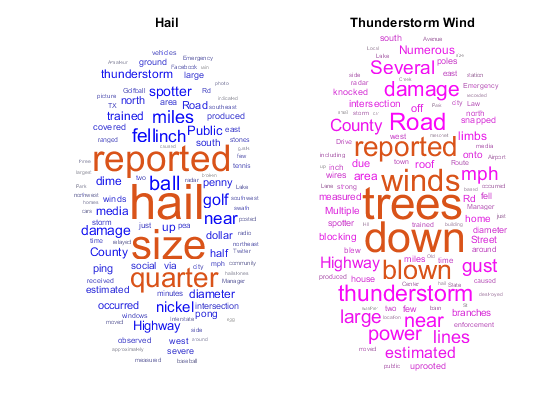
title("Hail")

subplot(1,2,2)

idx = labels == "Thunderstorm Wind";

wordcloud(textData(idx),'Color','magenta');

title("Thunderstorm Wind")



比较佛罗里达州, 堪萨斯州和阿拉斯加的报告中的单词。在矩形中为每个状态创建报告的单词云, 并在每个单词云周围绘制边框。

figure

state = T.state;

subplot(1,3,1)

idx = state == "FLORIDA";

wordcloud(textData(idx),'Shape','rectangle','Box','on');

title("Florida")

subplot(1,3,2)

idx = state == "KANSAS";

wordcloud(textData(idx),'Shape','rectangle','Box','on');

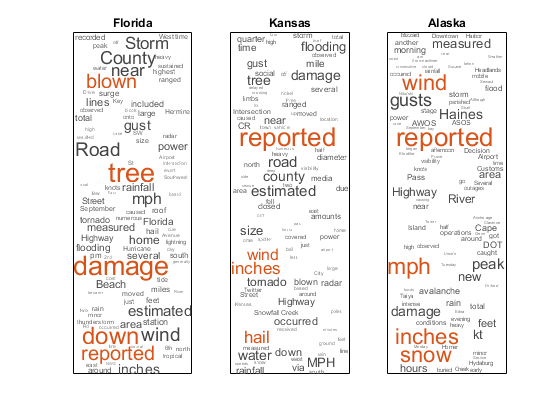
title("Kansas")

subplot(1,3,3)

idx = state == "ALASKA";

wordcloud(textData(idx),'Shape','rectangle','Box','on');

title("Alaska")



将报告中的文字与财产损失进行比较, 报告中以千美元计, 并以百万美元报告损坏报告。为每个这些金额创建报告的文字云, 并分别高亮显示蓝色和红色。

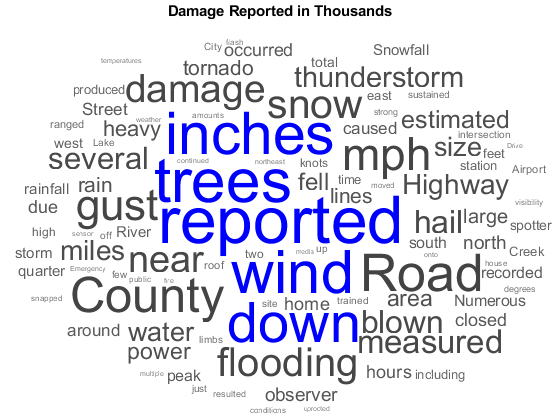
cost = T.damage\_property;

idx = endsWith(cost,"K");

figure

wordcloud(textData(idx),'HighlightColor','blue');

title("Damage Reported in Thousands")

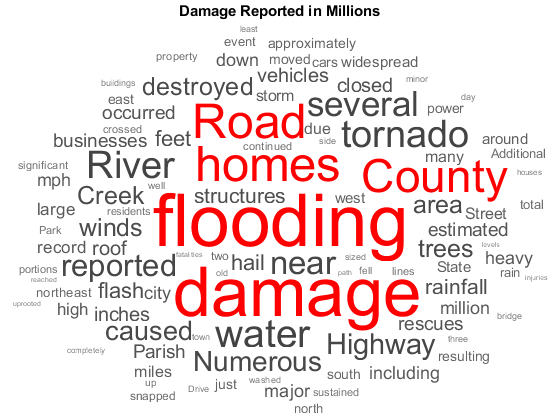


idx = endsWith(cost,"M");

figure

wordcloud(textData(idx),'HighlightColor','red');

title("Damage Reported in Millions")



## 使用文本散点图可视化 Word 嵌入

此示例演示如何使用2维和3维的嵌入和文本散点图来可视化 word。

单词嵌入, 将词汇中的单词映射到真实向量。向量试图捕获单词的语义, 以便类似的词具有相似的向量。一些嵌入也捕捉 "意大利是法国, 罗马是巴黎" 等词之间的关系。在向量形式中, 这种关系是https://www.mathworks.com/help/examples/textanalytics/win64/VisualizeWordEmbeddingsUsingTextScatterPlotsExample_eq01.png.

若要重现此示例中的结果, 请将rng设置为'default'.

rng('default')

### 加载预先训练字嵌入

使用fastTextWordEmbedding加载预先训练字嵌入。此函数需要 fastText 英语160亿令牌字嵌入支持包的文本分析 Toolbox™模型。如果未安装此支持包, 则该函数将提供下载链接。

emb = fastTextWordEmbedding

emb =

wordEmbedding with properties:

Dimension: 300

Vocabulary: [1×1000000 string]

探索使用word2vec和vec2word嵌入的单词。使用word2vec将意大利、 罗马和巴黎的词语转换为矢量.

italy = word2vec(emb,"Italy");

rome = word2vec(emb,"Rome");

paris = word2vec(emb,"Paris");

计算由italy - rome + paris提供的向量。这个向量封装了意大利语的语义意义, 没有罗马词的语义, 也包括了巴黎这个词的语义。.

vec = italy - rome + paris

vec = 1×300 single row vector

0.1606 -0.0690 0.1183 -0.0349 0.0672 0.0907 -0.1820 -0.0080 0.0320 -0.0936 -0.0329 -0.1548 0.1737 -0.0937 -0.1619 0.0777 -0.0843 0.0066 0.0600 -0.2059 -0.0268 0.1350 -0.0900 0.0314 0.0686 -0.0338 0.1841 0.1708 0.0276 0.0719 -0.1667 0.0231 0.0265 -0.1773 -0.1135 0.1018 -0.2339 0.1008 0.1057 -0.1118 0.2891 -0.0358 0.0911 -0.0958 -0.0184 0.0740 -0.1081 0.0826 0.0463 0.0043

使用vec2word在嵌入到vec中查找最接近的单词.

word = vec2word(emb,vec)

word =

"France"

### 创建2维文本散点图

使用tsne和textscatter创建2维文本散点图, 可视化嵌入的单词.

使用word2vec将前500个单词转换为矢量。V是长度为300的词向量的矩阵。

words = emb.Vocabulary(1:5000);

V = word2vec(emb,words);

size(V)

ans = 1×2

5000 300

使用tsne在二维空间中嵌入单词向量。此函数可能需要几分钟才能运行。如果要显示收敛信息, 请将'Verbose'名称-值对设置为1。

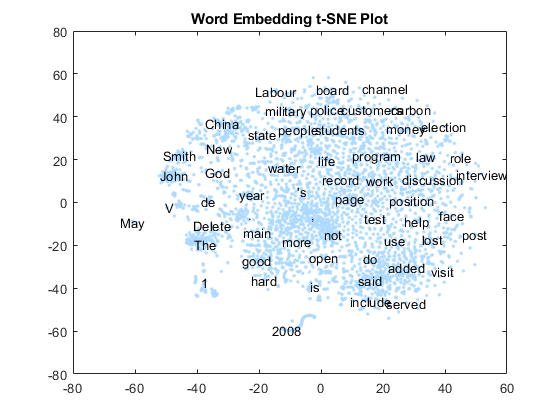
XY = tsne(V);

在2维文本散点图中按XY指定的坐标绘制单词。为便于阅读, 默认情况下, textscatter不显示所有输入字词, 而是显示标记。

figure

textscatter(XY,words)

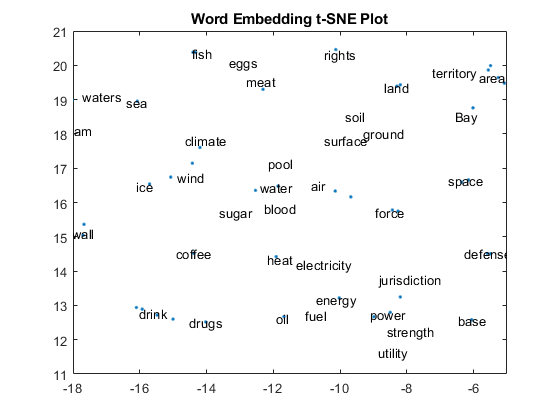
title("Word Embedding t-SNE Plot")



放大绘图的一部分。

xlim([-18 -5])

ylim([11 21])



### 创建3维文本散点图

使用tsne和textscatter创建3维文本散点图, 可视化嵌入的单词.

使用word2vec将前5000个单词转换为矢量。V是长度为300的词向量的矩阵。

words = emb.Vocabulary(1:5000);

V = word2vec(emb,words);

size(V)

ans = 1×2

5000 300

通过指定要为三的维数, 将单词向量嵌入到三维空间中使用tsne 。此函数可能需要几分钟才能运行。如果要显示收敛信息, 则可以将'Verbose'名称-值对设置为1。

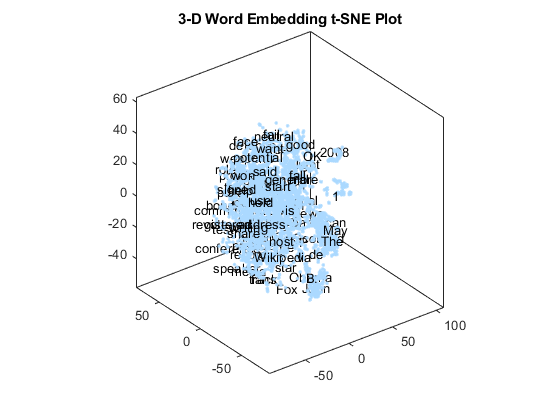
XYZ = tsne(V,'NumDimensions',3);

在3维文本散点图中按 XYZ 指定的坐标绘制单词。

figure

ts = textscatter3(XYZ,words);

title("3-D Word Embedding t-SNE Plot")

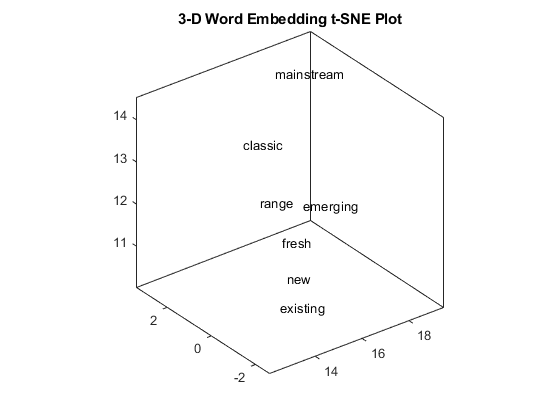


放大绘图的一部分。

xlim([12.04 19.48])

ylim([-2.66 3.40])

zlim([10.03 14.53])



### 执行群集分析

使用word2vec将前5000个单词转换为矢量。V是长度为300的词向量的矩阵。

words = emb.Vocabulary(1:5000);

V = word2vec(emb,words);

size(V)

ans = 1×2

5000 300

使用kmeans发现25集群.

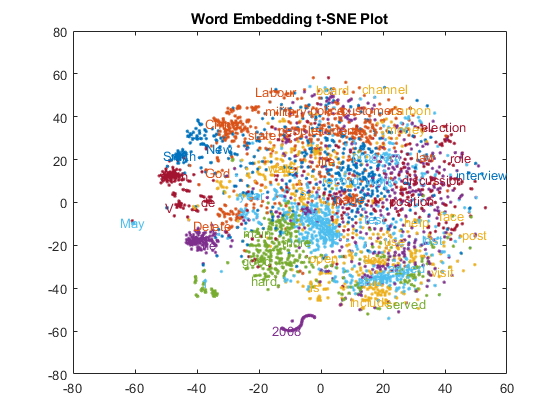
cidx = kmeans(V,25,'dist','sqeuclidean');

使用前面计算的2维 t-SNE 数据坐标, 在文本散点图中可视化簇。

figure

textscatter(XY,words,'ColorData',categorical(cidx));

title("Word Embedding t-SNE Plot")



放大绘图的一部分。

xlim([13 24])

ylim([-47 -35])

