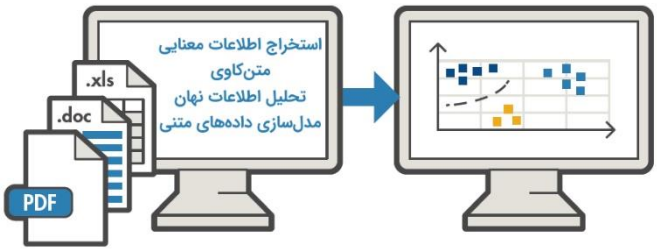
تولباکس تحلیل متن در متلب (Text Analytics Toolbox in MATLAB)

تولباکس تحلیل متن

تولباکس تحلیل (کمی و کیفی) متن، امکانات و ابزار مختلفی برای پیاده‌سازی الگوریتم‌های تحلیل متن و بصری‌سازی آن‌ها در اختیار کاربران قرار می‌دهد.

از این دسته از ابزارها، برای «پیش‌پردازش» (Pre-Process)، تحلیل و مدل‌سازی داده‌های متنی استفاده می‌شود.

از مدل‌های پیاده‌سازی شده توسط تولباکس تحلیل (کمی و کیفی) متن در کاربردهایی نظیر «تحلیل احساسات» (Sentiment Analysis)، «مدل‌سازی موضوعی» (Topic Modeling) و «پیش‌بینی شرایط نگه‌داری دستگاه‌ها» (Predictive Maintenance) استفاده می‌شود.



بصری‌سازی داده‌های متنی در متلب

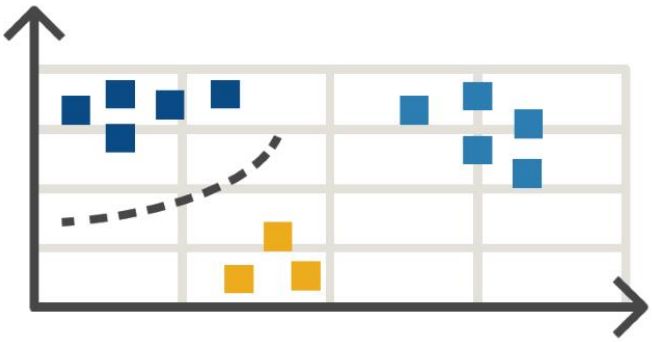
از نمودارهای تعریف شده با هدف تولید و نمایش «ابرهای کلمه» (Word Clouds) و «پراکندگی متنی» (Text Scatter)، جهت خلاصه‌سازی (Summarize) و صحت‌سنجی (Validate) نتایج حاصل از تحلیل (کمی و کیفی) داده‌های متنی استفاده می‌شود.

توابع لازم برای بصری‌سازی داده‌های متنی در متلب

نام تابع	توصیف تابع
`wordcloud`	این تابع، با استفاده از مدل‌هایی نظیر Bag-of-Words (BoW) و «تخصیص دیریکله نهان» (Latent Dirichlet allocation)، نمودارهای نمایش ابرِ کلمه‌ای را تولید می‌کند.
`wordCloudCounts`	این تابع، جهت تولید نمودارهای ابرِ کلمه، تعداد دفعات تکرار کلمات در داده‌های متنی را محاسبه می‌کند.

`textscatter`	این تابع، نمودار پراکندگی دوبُعدی (2D) را از داده‌های متنی تولید می‌کند.
`textscatter3`	این تابع، نمودار پراکندگی سه‌بُعدی (3D) را از داده‌های متنی تولید می‌کند.
`heatmap`	این تابع، نمودار «نقشه حرارت» (Heatmap) مرتبط با عناصر داده‌های متنی را تولید می‌کند.
`histcounts`	این تابع، داده متنی ورودی را از طریق الگوریتم‌های استاندارد به تعدادی «دسته» (Bin) تقسیم‌بندی می‌کند. در نهایت، اطلاعاتی نظیر تعداد عناصر در هر دسته و سایر موارد، به عنوان خروجی توسط تابع تولید می‌شود.
`discretize`	این تابع، داده‌های متنی (یا نمایش حاصل شده از آن‌ها) را در دسته‌ها یا طبقه‌های خاص گروه‌بندی می‌کند.

مدل‌سازی داده‌های متنی و پیش‌بینی در متلب



با استفاده از روش‌هایی نظیر Bag-of-Words و یا مدل‌های «تعبیه‌سازی کلمات» (Word Embedding) از پیش آموزش دیده، می‌توان داده‌های متنی را به «نمایش عددی» (Numerical Representation) متناظر آن‌ها تبدیل کرد.

همچنین، از طریق توابع تعریف شده در تولباکس تحلیل متن، می‌توان الگوریتم‌های یادگیری ماشین را برای پیش‌بینی و مدل‌سازی موضوعی مورد استفاده قرار داد.

توابع لازم برای مدل‌سازی داده‌های متنی و پیش‌بینی در متلب

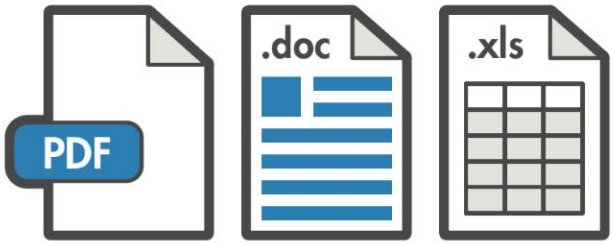
نام تابع	توصیف تابع
`readWordEmbedding`	با استفاده از این تابع، یک مدل «تعبیه‌سازی کلمات» (Word Embedding) از پیش آموزش دیده (که در فایل txt یا zip ذخیره شده است) توسط برنامه خوانده می‌شود.
`trainWordEmbedding`	با استفاده از این تابع، یک مدل «تعبیه‌سازی کلمات» (Word Embedding) آموزش داده می‌شود.
`word2vec/vec2word`	با استفاده از این تابع، کلمات موجود در داده‌های متنی به «بردارهای تعبیه‌سازی» (Embedding Vectors) نگاشت می‌شوند.
`ldaModel`	مدل «تخصیص دیریکله نهان» (Latent Dirichlet allocation) یا LDA
`lsaModel`	مدل «تحلیل معنای نهان» (Latent Semantic Analysis) یا LSA
`bagOfWords`	مدل Bag-of-Words یا BoW
`fitlda`	از این تابع، برای «برازش» (Fitting) مدل «تخصیص دیریکله نهان» (Latent Dirichlet allocation) یا LDA روی داده‌های متنی یا مدل نمایشی متناظر استفاده می‌شود.

برای مشاهده دیگر «تقلب‌نامه‌های» مجله فرادرس، به این لینک مراجعه فرمایید.

`fitlsa`	از این تابع، برای «برازش» (Fitting) مدل «تحلیل معنای نهان» (Latent Semantic Analysis) یا LSA روی داده‌های متنی یا مدل نمایشی متناظر استفاده می‌شود.
`predict`	از این مدل، جهت پیش‌بینی موضوعات برتر موجود در اسناد متنی با استفاده از مدل تخصیص دیریکله نهان یا LDA استفاده می‌شود.
`fitdist`	از این تابع، جهت برازش توزیع احتمالی روی داده‌های متنی استفاده می‌شود.
`fitrlinear`	از این تابع، برای برازش یک مدل «رگرسیون خطی» (Linear Regression) روی داده‌های با ابعاد بالا استفاده می‌شود.
`fitclinear`	از این تابع، برای برازش یک مدل «دسته‌بندی خطی» (Linear Classification) روی داده‌های با ابعاد بالا استفاده می‌شود.
`fitcecoc`	از این تابع، جهت برازش مدل‌های «چندکلاسه» (Multi-Class) برای دسته‌بندها (Classifiers) استفاده می‌شود.

خواندن اسناد و استخراج داده‌های متنی

با استفاده از توابع تعریف شده در تولباکس تحلیل متن، این امکان برای کاربران و برنامه‌نویسان فراهم شده است تا داده‌های متنی را از فایل‌های PDF، فایل‌های Microsoft Word، فایل‌های متنی ساده و فایل‌های «صفحه گسترده» (Spreadsheet) استخراج کنند.



توابع لازم برای خواندن اسناد و استخراج داده‌های متنی

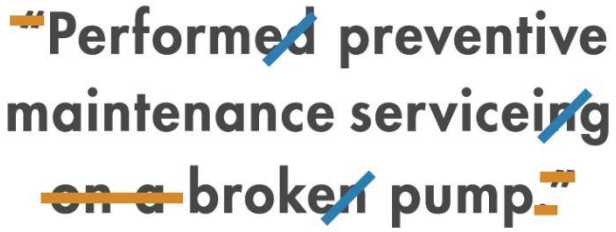
نام تابع	توصیف تابع
`extractFileText`	با استفاده از این تابع، داده‌های متنی از فایل‌های PDF، فایل‌های Microsoft Word و فایل‌های متنی ساده خوانده می‌شوند.
`textscan`	با استفاده از این تابع، «داده‌های متنی قالب‌بندی شده» (Formatted Text Data) از فایل‌ها یا رشته‌های متنی خوانده می‌شوند.
`readtable`	با استفاده از این تابع، ابتدا یک جدول خالی ساخته می‌شود. سپس، داده‌های متنی از فایل‌های «ستون محور» (Column-Oriented) خوانده می‌شوند. در نهایت، محتویات خوانده شده، در جدول تولید شده ذخیره می‌شوند.
`compose`	از این تابع، برای تبدیل داده‌های متنی به آرایه‌های قالب‌بندی شده از نوع «رشته» (String) استفاده می‌شود.

`xlsread`	از این تابع، برای خواندن محتویات موجود در فایل‌های صفحه گسترده Microsoft Excel استفاده می‌شود.
`webread`	از این تابع، برای خواندن محتویات از سرویس‌های تحت وب RESTful استفاده می‌شود.
`FileDatastore`	از این تابع، برای ساختن یک datastore استفاده می‌شود. در متلب، یک datastore مخزنی متشکل از مجموعه‌های داده‌ای بسیار بزرگ است که نمی‌توانند در حافظه جای بگیرند. یک datastore، به کاربران و برنامه‌نویسان اجازه می‌دهد تا تمامی داده‌های خوانده و پردازش شده را در قالب یک موجودیت واحد ذخیره‌سازی و گردآوری کنند.
`TabularTextDatastore`	از این تابع، جهت ساختن datastore برای فایل‌های متنی جدولی (Tabular Text Files) استفاده می‌شود.
`SpreadsheetDatastore`	از این تابع، جهت ساختن datastore برای فایل‌های صفحه گسترده (Spreadsheet Files) استفاده می‌شود.

پیش‌پردازش داده‌های متنی

توابع تعریف شده در تولباکس تحلیل متن متلب، این امکان را برای کاربران این زبان برنامه‌نویسی فراهم می‌آورند تا «ویژگی‌هایی» (Features) را که بار معنایی ندارند و برای تحلیل (کمی و کیفی) داده‌های متنی مفید نیستند، شناسایی و حذف کنند.

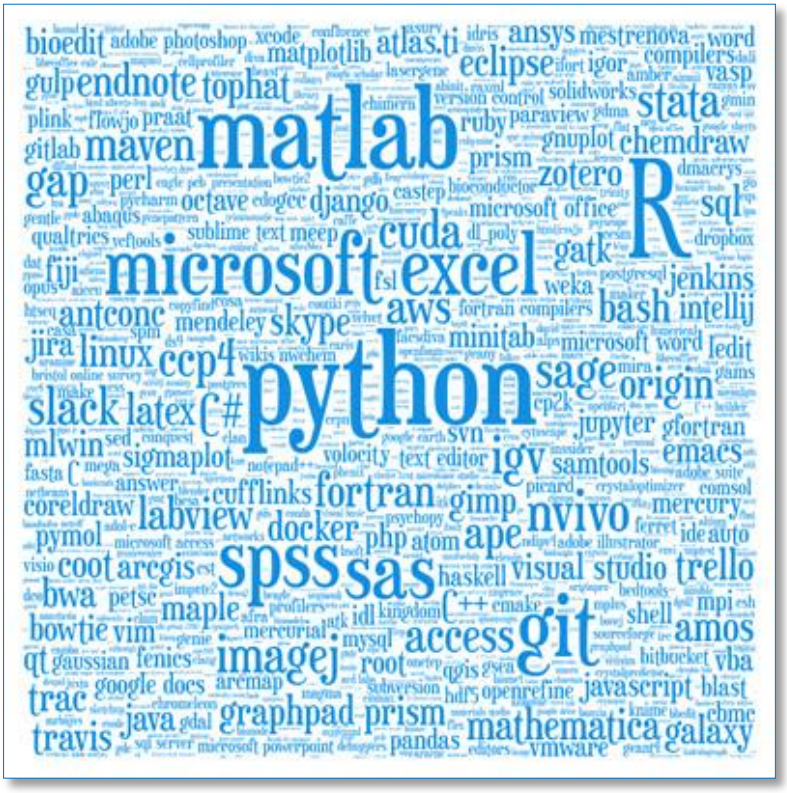
ویژگی‌هایی نظیر کلمات شایع، علائم نقطه‌گذاری و آدرس‌های اینترنتی از جمله چنین ویژگی‌هایی هستند. همچنین، تکنیک‌هایی برای «نرمال‌سازی داده‌های متنی» (Text Normalization) در تولباکس تحلیل متن تعریف شده‌اند که برای بازگرداندن کلمات به «فرم ریشه‌ای» (Root Form) مورد استفاده قرار می‌گیرند (فرایند Stemming).



توابع لازم برای پیش‌پردازش داده‌های متنی

نام تابع	توصیف تابع
`tokenizedDocument`	از این تابع، برای تقسیم‌بندی داده‌ها و اسناد متنی به مجموعه‌ای از کلمات استفاده می‌شود.
`normalizeWords`	در این تابع، از الگوریتم شناخته شده Porter جهت انجام عملیات Stemming و برگرداندن کلمات به شکل ریشه‌ای استفاده می‌شود.
`bagOfWords`	از این تابع، برای مدل‌سازی Bag-of-Words استفاده می‌شود.

در این لیست، مجموعه «کلمات بی‌اثر» (Stopwords) در زبان‌های مختلف سازمان‌دهی شده‌اند. کلمات بی‌اثر در زبان‌های مختلف، کلماتی هستند که بار معنایی ندارند و نقشی در تعیین قالب محتوایی و زمینه موضوعی اسناد ایفا نمی‌کنند.	`stopWords`
از این تابع، برای جستجوی ظاهر شدن یک کلمه خاص در اسناد متنی استفاده می‌شود. با استفاده از این تابع، علاوه بر آمار مرتبط با ظاهر شدن یک کلمه در اسناد متنی، «زمینه موضوعی» (Context) ظاهر شدن کلمه نیز استخراج می‌شود.	`context`
با استفاده از این تابع، مجموعه‌ای از کلمات انتخابی (نظیر کلمات بی‌اثر) از اسناد متنی یا مدل‌های Bag-of-Words حذف می‌شوند.	`removeWords`
با استفاده از این تابع، کلمات بلند (با تعداد کاراکترهای زیاد) از اسناد متنی یا مدل‌های Bag-of-Words حذف می‌شوند. پارامتر طول (تعداد کاراکترها) کلمات توسط کاربر مشخص می‌شود.	`removeLongWords`
با استفاده از این تابع، کلمات کوتاه (با تعداد کاراکترهای کم) از اسناد متنی یا مدل‌های Bag-of-Words حذف می‌شوند. پارامتر طول کلمات (تعداد کاراکترها) توسط کاربر مشخص می‌شود.	`removeShortWords`
با استفاده از این تابع، کلماتی که تعداد دفعات تکرار آن‌ها در اسناد یا داده‌های متنی یا مدل Bag-of-Words کم است، حذف می‌شوند. پارامتر تعداد دفعات تکرار مطلوب، توسط کاربر مشخص می‌شود.	`removeInfrequentWords`
با استفاده از این تابع، علائم نقطه‌گذاری از اسناد و داده‌های متنی حذف می‌شوند.	`erasePunctuation`



برای مشاهده دیگر «تقلب‌نامه‌های» مجله فرادرس، به این لینک مراجعه فرمایید.

نوع داده «رشته» (String)

توابع تعریف شده در تولباکس تحلیل متن، این امکان را برای کاربران و برنامه‌نویسان فراهم می‌آورند تا داده‌های متنی را به شکل بهینه‌ای «دستکاری» (Manipulate)، «مقایسه» (Compare) و «ذخیره» (Store) کنند.

```
"Hello,world"
```

توابع لازم برای دستکاری، مقایسه و ذخیره متغیرهای نوع داده «رشته» (String)

نام تابع	توصیف تابع
`str = "Hello,world"`	این دستور، چگونگی تعریف متغیر رشته‌ای (یک متغیر از نوع داده رشته) را نمایش می‌دهد.
`str = ["Hello", "World"]`	این دستور، چگونگی تعریف آرایه رشته‌ای (یک آرایه از نوع داده رشته) را نمایش می‌دهد.
`str = string( C )`	از این تابع، برای تبدیل یک بردار از کاراکترها به نام C، به یک متغیر رشته‌ای استفاده می‌شود.
`str2double`	از این تابع، برای تبدیل یک متغیر رشته‌ای به اعداد اعشاری (از نوع double) استفاده می‌شود.
`strlength`	با استفاده از این تابع، طول رشته‌ها در خروجی نمایش داده می‌شود.
`isstring`	این تابع، آرگومان ورودی را مورد بررسی قرار می‌دهد تا مشخص کند که این ورودی، آرایه‌ای از نوع رشته است یا نه.
`join`	از این تابع، برای ترکیب رشته‌ها استفاده می‌شود.
`split`	از این تابع، برای تقسیم‌بندی رشته‌ها به آرایه رشته‌ای استفاده می‌شود.
`splitlines`	در این تابع، جهت تقسیم‌بندی رشته‌ها به آرایه رشته‌ای، باید کاراکترهای newline در رشته مشاهده شوند.
`replace`	از این تابع، برای پیدا و جا به جا کردن «زیر رشته‌ها» (Substring) در یک آرایه رشته‌ای استفاده می‌شود.
`contains`	با استفاده از این تابع، رشته‌ای که به عنوان آرگومان ورودی به تابع پاس داده می‌شود مورد بررسی قرار می‌گیرد تا مشخص شود یک زیر رشته یا «الگوی» (Pattern) خاص در این رشته وجود دارد یا نه.
`erase`	از این تابع، برای حذف کردن زیر رشته‌های درون متغیرهای رشته‌ای استفاده می‌شود.
`extractBetween`	از این تابع، برای استخراج زیر رشته‌های موجود میان «شاخص‌های» (Indicators) تعریف شده استفاده می‌شود.

برای مشاهده دیگر «تقلب‌نامه‌های» مجله فرادرس، به این لینک مراجعه فرمایید.

`extractAfter`	از این تابع، برای استخراج زیر رشته از یک مکان خاص به بعد، در یک متغیر رشته‌ای استفاده می‌شود.
`extractBefore`	از این تابع، برای استخراج زیر رشته تا یک مکان خاص، در یک متغیر رشته‌ای استفاده می‌شود.
`strcmp`	از این تابع، برای مقایسه محتویات متغیرهای رشته‌ای استفاده می‌شود.
`regexp`	از این تابع، برای انجام عملیات مرتبط با «عبارات منظم» (Regular Expressions) استفاده می‌شود؛ به طور ویژه، برای تطبیق دادن عبارات منظم با رشته‌ها و زیر رشته‌های موجود در آن‌ها. شایان ذکر است که عملیات عبارات منظم روی رشته‌ها در زبان متلب، به کوچک یا بزرگ بودن حروف حساس است (Case Sensitive).

منبع