

Spectrum Technology Platform

バージョン 12SP1

Information Extraction ガイド



目次

Entity Extractor	34
Text Categorizer	37
Relationship Extractor	39

1 - はじめに

Information Extraction モジュール	4
サポートされている言語	4
モデル セキュリティ	4

2 - テキスト分類

テキスト分類の概要	6
データの準備	6
オプションの設定	7
モデルのトレーニング	11
モデルの評価	11
テキストの分類	11

3 - エンティティ抽出

はじめに	14
既存エンティティ	14
カスタム エンティティ	15

4 - 関連性抽出

はじめに	23
関連性タイプ	24

5 - ステージ リファレンス

Information Extraction コンポーネント	29
Read from Documents	29

1 - はじめに

このセクションの構成

Information Extraction モジュール	4
サポートされている言語	4
モデル セキュリティ	4

Information Extraction モジュール

Information Extraction モジュールは、高度なテキスト処理機能を提供し、自然言語で入力されたテキストから情報を抽出することができます。

サポートされている言語

最新のリリースの段階で、Information Extractionモジュールのすべてのステージにおいて情報抽出機能がサポートするのは英語の入力テキストのみです。

注： **Entity Extractor** ステージについては、英語以外に次の言語サポートがベータ版の段階にあります。

es	スペイン語 (メキシコ)
fr	フランス語
de	ドイツ語
pt	ポルトガル語 (ブラジル)

重要：これらのベータ版言語は、**Custom Entity** の場合のみ使用可能で、既存エンティティでは使用できません。

モデル セキュリティ

Information Extractionを使用して各種機能を実行するには、**Management Console** でセキュリティ権限を割り当てる必要があります。

- モデルの分類または一覧表示には、表示権限が必要です。
- モデルの再トレーニングまたはインポートには、更新権限が必要です (モデルが既に存在する場合)。
- モデルのインポートまたはトレーニングには、作成権限が必要です。
- モデルの削除には、削除権限が必要です。

2 - テキスト分類

このセクションの構成

テキスト分類の概要	6
データの準備	6
オプションの設定	7
モデルのトレーニング	11
モデルの評価	11
テキストの分類	11

テキスト分類の概要

テキスト分類 (テキストのカテゴリ化とも呼ばれます) は、構造化されていないコンテンツまたはプレーン テキスト (電子メール、ニュース記事、コメントなど) にカスタム カテゴリを割り当てる処理です。コンテンツは、題名、著者、日付、または、ユーザが考案した実質的に任意の分類体系に基づいて分類できます。

カテゴリライザ モデルにお使いのデータやカテゴリを学習させることによって、お客様独自のカテゴリライザを作成することができます。まず、トレーナーがデータを分析し、トレーニング プロセスで取得した情報を保存します。次にコンテンツを分析し、コンテンツが属するカテゴリを決定します。

テキスト分類機能は、統計的なテキスト分類プロセスを使用します。このプロセスでは、人間が作成したトレーニング ドキュメントに基づく自動分類ルールを学習するための、機械学習手法を適用します。

自由に選択した分類方法が適用できるため、最初にそのカテゴリを「学習」するようにモデルを「トレーニング」する必要があります。それを終えてから、モデルを **Text Categorizer** ステージで使用して、構造化されていないデータを分類することができます。

Spectrum™ Technology Platformでは、管理ユーティリティ コマンドを使用して、テキスト分類モデルを管理します。コマンドの説明については、[管理ユーティリティ コマンド](#)を参照してください。

データの準備

テキスト分類を使用するための最初のステップは、入力ファイルとテスト ファイルを準備することです。そのためには、両方のファイル内で、タブで区切られた値としてデータを構造化する必要があります。ファイルには、次の形式で詳細情報が記述されている必要があります。

- UFT-8 エンコーディング
- 2 行のタブ区切りデータ。1 行目にはカテゴリ名 ("Patient" や "Provider" など)、2 行目には各カテゴリのデータ (以下の例を参照してください) が含まれます。

データは次のようになります。

```
Patient      John Smith dob04181963 224 Main St. Atl GA 30311
Provider     Mark Johnson M.D. NPI5489512047 412 Washington Atl GA 30301
```

オプションの設定

ここでは、トレーニング オプション ファイルを作成する必要があります。このファイルには、モデルに関する情報と、モデルのトレーニングに適用するオプションが含まれます。このファイルは、UTF-8エンコーディングのXML形式で、以下に示すヘッダと必須トレーニング機能を含む必要があります。

トレーニング オプション ファイル内のヘッダ

ヘッダには、モデルの詳細情報、タイプ、テストおよび入力ファイルのパスが記述されます。

- modelName: モデル名
- modelType: モデルのタイプ (ここではテキスト分類を意味する TC)
- modelDescription: モデルの説明
- inputFilePath: モデルのトレーニングに使用する入力ファイルの場所
- testFilePath: テスト ファイルの場所

注:

テスト ファイルは、モデルの有効性を評価します。さまざまなトレーニング パラメータによるカスタム モデルの動作を判定します。ベスト プラクティスとして、カスタム モデルのトレーニングまたは評価には、異なる入力およびテスト ファイルを使用することを推奨します。

algorithm: モデルのトレーニングに使用する機械学習アルゴリズム (デフォルトは MaxEnt)

トレーニング機能

以下のトレーニング機能を使用して、新しいカテゴリを作成できます。

注: 複数の機能を使用する場合、ファイル内でのそれらの順序は任意です。

- **言語機能:** 言語プロパティを指定します。
 - Stemming: 単語を語幹部分に短縮します。例えば、"insurer"、"insured"、"insures" はすべて、語幹である "insure" に短縮できます。

```
<trainingFeature>
  <featureName>Stemming</featureName>
</trainingFeature>
```

- **キーワード機能:** キーワードのリストを定義します。

- **IgnoreWords**: ストップワードとも呼ばれるこの機能は、"the"、"and"、"but" など、分類に影響を与えない一般的な単語を除外します。これらの単語は、空白を入れずにカンマのみで区切る必要があります。この機能では Append キーも使用できます。これを "True" に設定すると、ストップワードの既存リストに追加されます。

```
<trainingFeature>
  <featureName>IgnoreWords</featureName>
  <featureParams>
    <entry>
      <key>WordList</key>
      <value>
        and,the,for,with,still,tri,rep,cust,keep,get,req,call
      </value>
    </entry>
    <entry>
      <key>Append</key>
      <value>True</value>
    </entry>
  </featureParams>
</trainingFeature>
```

- **CategoryKeywords**: 複数のカスタム リストに属するキーワードのリストのカテゴリを識別します。例えば、CategoryKeywords リストの **Weekdays** には、Monday、Tuesday、Wednesday、Thursday、および Friday という Keywords が含まれます。

この機能では、マッチングで大文字と小文字を区別するかどうかも指定できます。デフォルトは true(大文字と小文字を区別) です。

```
<trainingFeature>
  <featureName>CategoryKeywords</featureName>
  <featureParams>
    <entry>
      <key>Weekdays</key>
      <!-- List of weekdays -->
      <value>Monday,Tuesday,Wednesday,Thursday,Friday</value>
    </entry>
    <entry>
      <key>WeekendDays</key>
      <!-- List of weekend days -->
      <value>Saturday,Sunday</value>
    </entry>
    <entry>
      <key>CaseSensitive</key>
      <value>True</value>
    </entry>
  </featureParams>
</trainingFeature>
```


- KeyWords: *DaysOfWeek* や *Month* など、ユーザが指定したカスタム リストに属する単語を検索します。大文字と小文字を区別して検索するかどうかも指定できます。デフォルトは "true" (大文字と小文字を区別) です。

```
<trainingFeature>
  <featureName>KeyWords</featureName>
  <featureParams>
    <entry>
      <key>KeyWordList</key>
      <value>Monday, Tuesday</value>
    </entry>
    <entry>
      <key>CaseSensitive</key>
      <value>False</value>
    </entry>
  </featureParams>
</trainingFeature>
```

- **語彙機能:** 語彙プロパティを指定します。
- NGram: 長い文字列の一部を検索します。"n" は検索する単語の数を表します。例えば、"to be or not to be" というフレーズを探している場合に、"to" または "be" というユニグラム、"to be" または "or not" というバイグラム、"to be or" または "not to be" というトライグラムを検索できます。

```
<trainingFeature>
  <featureName>NGram</featureName>
  <featureParams>
    <entry>
      <key>Count</key>
      <value>3</value>
    </entry>
  </featureParams>
</trainingFeature>
```

トレーニング オプション ファイルのサンプルを以下に示します。

```
<trainingOptions>
  <modelName>modelone</modelName>
  <modelType>TC</modelType>
  <modelDescription>modelOne</modelDescription>

  <inputFilePath>C:/SpectrumIE/textclassification/train_Input.csv</inputFilePath>

  <testFilePath>C:/SpectrumIE/textclassification/train_Test.txt</testFilePath>

  <algorithm>SVM</algorithm>
```

```
<trainingFeatures>

<!-- Keyword features -->
<trainingFeature>
  <featureName>IgnoreWords</featureName>
  <featureParams>
    <entry>
      <key>WordList</key>
      <value>
        and, the, for, with, still, tri, rep, cust, keep, get, req, call
      </value>
    </entry>
    <entry>
      <key>Append</key>
      <value>True</value>
    </entry>
  </featureParams>
</trainingFeature>

<trainingFeature>
  <featureName>CategoryKeywords</featureName>
  <featureParams>
    <entry>
      <key>Category1</key>
      <value>CategoryKeyword1, CategoryKeyword2</value>
    </entry>
    <entry>
      <key>Category2</key>
      <value>CategoryKeyword3, CategoryKeyword4</value>
    </entry>
  </featureParams>
</trainingFeature>

<trainingFeature>
  <featureName>KeyWords</featureName>
  <featureParams>
    <entry>
      <key>KeyWordList</key>
      <value>
        jam, misfeed, install, help, mechanical, failure, jam, pc, connection
      </value>
    </entry>
  </featureParams>
</trainingFeature>

<!-- Linguistic feature -->
<trainingFeature>
  <featureName>Stemming</featureName>
</trainingFeature>

<!-- Lexical feature -->
<trainingFeature>
```

```
<featureName>NGram</featureName>
<featureParams>
  <entry>
    <key>Count</key>
    <value>3</value>
  </entry>
</featureParams>
</trainingFeature>

</trainingFeatures>
</trainingOptions>
```

モデルのトレーニング

オプションファイルを作成した後は、予測される関連性を検出するようにモデルをトレーニングする必要があります。機械学習手法を適用することによって、これを行います。Spectrum™ Technology Platformでは、**iemodel train**のCLIコマンドを使用してモデルのトレーニングを行います。トレーニングを終えたモデルは、分類に使用できます。

モデルの評価

モデルをトレーニングした後はテストを行い、トレーニングオプションファイルが正しく、カテゴリが期待どおりに割り当てられることを確認するとよいでしょう。

モデルのテストには、**iemodel trainAndevaluate model**のCLIコマンドが使用できます。

テキストの分類

1. **Read from File** や **Read from XML** などのソース ステージ、**Text Categorizer** ステージ、そして **Write to File** や **Write to XML** などのシンク ステージが含まれるデータフローを作成します。
2. ソース ステージで、入力ファイルを指定します。

3. **Text Categorizer** ステージで、**[カテゴリライザ名]** フィールドでモデルを選択します。これは、テキスト分類フェーズでトレーニングしたモデルです。モデルのトレーニングについては、[モデルのトレーニング](#) (11ページ) を参照してください。
4. **[カテゴリ数]** フィールドで、出力に含めるカテゴリのマッチ レベルを選択します。例えば、最も近い一致、最も近いものと 2 番目に近いもの、などを選択できます。

注: 最大値は、モデルのトレーニング中に指定した異なるカテゴリの数に対応します。

5. **[OK]** をクリックします。
6. ジョブを実行します。

3 - エンティティ抽出

このセクションの構成

はじめに	14
既存エンティティ	14
カスタム エンティティ	15

はじめに

エンティティ抽出とは、構造化されていないデータからエンティティを識別して取得する処理です。**Entity Extractor** ステージに含まれる既存のエンティティを使用するか、カスタム エンティティを抽出するようにモデルをトレーニングすることができます。

既存エンティティ

既存エンティティとは、**Information Extraction** モジュールにあらかじめ用意されているエンティティのことです。

既存のエンティティを一覧表示するには、**Entity Extractor** ステージを開き、**[次の値でシステム デフォルト オプションを上書きします]** チェック ボックスを選択し、**[クイック追加]** をクリックします。エンティティのリストが **[エンティティの選択]** セクションに表示されます。

- *Person*
- *Address*
- *ProperNouns*
- *ISBN*
- *CreditCard*
- *ZipCode*
- *WebAddress*
- *Mention*
- *HashTag*
- *SSN*
- *Phone*
- *Email*
- *Date*
- *Location*
- *Organization*

このセクションの残りの手順に従うことにより、これらの種類のエンティティをデータから抽出することができます。

既存エンティティの抽出

1. **Read from Documents** ソース ステージ、**Entity Extractor** ステージ、およびシンク ステージ (**Write to File** や **Write to XML** など) を含む、データフローを作成します。
2. ソース ステージで、入力ファイルを指定します。
3. **Entity Extractor** ステージで、入力文字列から抽出するデータに基づいてエンティティを選択します。例えば、ファイルに含まれているすべての人名と住所を選択するなら、**Address**と **Person** エンティティを選択します。

注: **Address**と**Person** はデフォルトのエンティティです。他のエンティティに基づいてデータを抽出するには、**[次の値でシステム デフォルト オプションを上書きします]** チェック ボックスを選択し、**[クイック追加]** をクリックします。エンティティのリストが **[エンティティの選択]** セクションに表示されます。

4. 指定したエンティティに関連するデータの入力ファイル内での出現回数を取得するには、**[出力エンティティ数]** チェック ボックスを選択します。
5. **[OK]** をクリックします。
6. ジョブを実行します。

カスタム エンティティ

既存のエンティティと同じように、カスタム エンティティを取得するようにモデルをトレーニングできます。これらのエンティティは、任意のドメインに属することができ、任意のタイプとすることができます。例えば、医療関連のテキストを用いて、診断や医薬品のリストを抽出することができます。カスタム エンティティの抽出は、次の手順で実行します。

1. データの準備: 入力ファイルとテスト ファイルの準備
2. オプションの設定: モデルに関する情報と、モデルのトレーニングに適用するオプションを含む、トレーニング オプション ファイルの作成
3. モデルのトレーニング
4. エンティティの抽出

これらの手順をすべて正常に実行すると、新しいエンティティ タイプが **Entity Extractor** ステージのリストに追加され、それを使用して構造化されていないファイルから情報を抽出することができます。

カスタム エンティティ用のデータの準備

カスタムエンティティを作成するための最初のステップは、入力ファイルとテストファイルを準備することです。カスタム エンティティ機能を利用するには、これらのファイル内のエンティティを、トレーニング オプション ファイルで指定した `magicWord` によって囲む必要があります (これについては次のトピックを参照してください)。

例えば、入力ファイル内の構造化されていないデータから診断結果を抽出する場合を考えます。トレーニング オプション ファイルでは、`magicWordDIAGNOSIS`を指定したとします。テキスト内に出現するすべての病気や症状の名称に対し、次のようにその語をその`magicWord`で囲む必要があります。

```
The term diagnostic criteria designates the specific combination of signs, symptoms, and test results that the clinician uses to attempt to determine the correct diagnosis. Some examples of diagnostic criteria, also known as clinical case definitions, are: Amsterdam criteria for DIAGNOSIShereditary nonpolyposis colorectal cancerDIAGNOSIS McDonald criteria for DIAGNOSISmultiple sclerosisDIAGNOSIS ACR criteria for DIAGNOSISsystemic lupus erythematosusDIAGNOSIS Centor criteria for DIAGNOSISstrep throatDIAGNOSIS.
```

`magicWord` の識別については、次のトピックを参照してください。

カスタム エンティティのオプションの設定

ここでは、トレーニング オプション ファイルを作成する必要があります。このファイルには、モデルに関する情報と、モデルのトレーニングに適用するオプションが含まれます。このファイルは、UFT-8エンコーディングのXML形式で、以下に示すヘッダと必須トレーニング機能を含む必要があります。

トレーニング オプション ファイル内のヘッダ

ヘッダには、モデルの詳細情報、テストおよび入力ファイルのパス、そしてカスタム エンティティに注釈を付けるためのキーワードが記述されます。

- `modelName`: カスタム モデルの名前
- `modelType`: カスタム モデルのタイプ (*CustomEntity*)
- `modelDescription`: カスタム モデルの説明
- `inputFilePath`: モデルのトレーニングに使用するタグ付きファイル (入力ファイル) のパス

- testFilePath: モデルのテストに使用するファイルのパス
- magicWord: カスタム エンティティの注釈付けに使用するキーワード
- language: テキストに使用される言語。

注: 英語がサポートされています。オランダ語、フランス語、ドイツ語、スペイン語がベータ版の段階にあります。

トレーニング機能

以下のトレーニング機能を使用して、カスタム エンティティを作成できます。

- **言語機能:** 言語プロパティを指定します。
 - POSTagger: 名詞、代名詞、形容詞、動詞など、言語の部分を識別するためのタグ付けを行います。

```
<trainingFeature>
  <featureName>POSTagger</featureName>
</trainingFeature>
```

- **表記機能:** 構造的プロパティを指定します。
 - CaseIdentifier: カスタム エンティティがすべて大文字か、すべて小文字か、大文字と小文字が混合しているかを識別します。

```
<trainingFeature>
  <featureName>CaseIdentifier</featureName>
</trainingFeature>
```

- NumericIdentifier: カスタム エンティティが数字か英数字かを識別します。

```
<trainingFeature>
  <featureName>NumericIdentifier</featureName>
</trainingFeature>
```

- 1st2ndIdentifier: カスタム エンティティが 1st、2nd、3rd などの序数かどうかを識別します。

```
<trainingFeature>
  <featureName>1st2ndIdentifier</featureName>
</trainingFeature>
```

- **PatternMatcher**: 正規表現を用いて1つ以上のパターンに対して単語をマッチングします。複数の正規表現を指定する場合は、ANDか、いずれかが成立すればよい OR (デフォルト) を含めます。

```
<trainingFeature>
  <featureName>PatternMatcher</featureName>
  <featureParams>
    <entry>
      <key>RegEx1</key>
      <value>b[aeiou]t</value>
    </entry>
    <entry>
      <key>RegEx2</key>
      <value>b[xyz]t</value>
    </entry>
    <entry>
      <key>JoinCondition</key>
      <value>AND</value>
    </entry>
  </featureParams>
</trainingFeature>
```

- **キーワード機能**: キーワードのリストを定義します。
- **CategoryKeywords**: 複数のカスタム リストに属するキーワードのリストのカテゴリを識別します。例えば、CategoryKeywords リストの **Weekdays** には、**Monday**、**Tuesday**、**Wednesday**、**Thursday**、および **Friday** という Keywords が含まれます。

この機能では、マッチングで大文字と小文字を区別するかどうかも指定できます。デフォルトは true(大文字と小文字を区別) です。

```
<trainingFeature>
  <featureName>CategoryKeywords</featureName>
  <featureParams>
    <entry>
      <key>Weekdays</key>
      <!-- List of weekdays -->
      <value>Monday, Tuesday, Wednesday, Thursday, Friday</value>
    </entry>
    <entry>
      <key>WeekendDays</key>
      <!-- List of weekend days -->
      <value>Saturday, Sunday</value>
    </entry>
    <entry>
      <key>CaseSensitive</key>
      <value>True</value>
    </entry>
  </featureParams>
</trainingFeature>
```

- KeyWords: *DaysOfWeek* や *Month* など、ユーザが指定したカスタム リストに属する単語を検索します。大文字と小文字を区別して検索するかどうかも指定できます。デフォルトは "true" (大文字と小文字を区別) です。

```
<trainingFeature>
  <featureName>KeyWords</featureName>
  <featureParams>
    <entry>
      <key>KeyWordList</key>
      <value>Monday, Tuesday</value>
    </entry>
    <entry>
      <key>CaseSensitive</key>
      <value>False</value>
    </entry>
  </featureParams>
</trainingFeature>
```

- Substring: パラメータで指定された部分文字列を抽出します。接頭語と接尾語の抽出にも使用できます。
 - StartLocation: **Left** (左) または **Right** (右)。文字列を抽出する位置。デフォルトは Left です。
 - StartPosition: 部分文字列の開始位置。デフォルトは 0 です。
 - EndPosition: 部分文字列の終了位置。デフォルトは 3 です。
 - MinLength: この機能を適用する単語の最小の長さ。デフォルトは 3 です。

```
<trainingFeature>
  <featureName>Substring</featureName>
  <featureParams>
    <entry>
      <key>StartLocation</key>
    </entry>
    <entry>
      <key>StartPosition</key>
      <value>1</value>
    </entry>
    <entry>
      <key>EndPosition</key>
      <value>4</value>
    </entry>
    <entry>
      <key>MinLength</key>
    </entry>
  </featureParams>
</trainingFeature>
```

- **語彙機能**: 語彙プロパティを指定します。

- FeatureWindow: 機能生成のウィンドウを指定します。

```
<trainingFeature>
  <featureName>FeatureWindow</featureName>
  <!-- Number of preceding tokens used to create the feature set.
Default is 3 -->
  <entry>
    <key>Before</key>
    <value>1</value>
  </entry>
  <!-- Number of succeeding tokens used to create the feature set.
Default is 3 -->
  <entry>
    <key>After</key>
    <value>2</value>
  </entry>
</trainingFeature>
```

カスタム エンティティに対する、完全なサンプル トレーニング オプション ファイルのサンプルを以下に示します。

```
<trainingOptions>
  <modelName>CustomModel</modelName>
  <modelType>CustomEntity</modelType>
  <modelDescription>CustomDiagnosesModel</modelDescription>

  <inputFilePath>C:/SpectrumIE/custom_model/Custom_Input.csv</inputFilePath>

  <testFilePath>C:/SpectrumIE/custom_model/Custom_Test.txt</testFilePath>

  <magicWord>DIAGNOSIS</magicWord>
  <language>English</language>

  <trainingFeatures>

  <!-- Lexical features-->
  <trainingFeature>
    <featureName>FeatureWindow</featureName>
    <featureParams>
      <entry>
        <key>Before</key>
        <value>1</value>
      </entry>
      <entry>
        <key>After</key>
        <value>2</value>
      </entry>
    </featureParams>
  </trainingFeature>

  <!-- Orthographic features-->
```

```
<trainingFeature>
  <featureName>CaseIdentifier</featureName>
</trainingFeature>

<trainingFeature>
  <featureName>NumericIdentifier</featureName>
</trainingFeature>
</trainingFeatures>
</trainingOptions>
```

カスタム エンティティ モデルのトレーニング

オプションファイルを作成した後は、カスタム エンティティを識別するようにモデルをトレーニングする必要があります。Spectrum™ Technology Platformでは、**iemodel train**の CLI コマンドを使用してこれを行います。トレーニング済みモデルを使用して、カスタム エンティティを取得します。

カスタム エンティティ モデルの評価

モデルをトレーニングした後はテストを行い、トレーニングオプションファイルが正しく、エンティティが期待どおりに抽出されることを確認するとよいでしょう。モデルをテストするには、**iemodel trainAndevaluate model**の CLI コマンドを使用します。

カスタム エンティティの抽出

トレーニング済みのカスタム エンティティが **Entity Extractor** ステージのエンティティ リストに追加され、構造化されていないデータから関連する情報を抽出できるようになりました。

既存のエンティティを抽出する方法については、[既存エンティティの抽出](#)（15ページ）を参照してください。

4 - 関連性抽出

このセクションの構成

はじめに	23
関連性タイプ	24

はじめに

Relationship Extraction を使用すると、ソース コンテンツ内で識別されたエンティティのペア間に成立する関連性タイプを識別できます。ソース ドキュメントの自然言語コンテンツを分析し、識別されたすべてのエンティティ ペアについて関連性タイプを識別します。

サポートされるエンティティ タイプ

現在、関連性の抽出機能でサポートされているエンティティ タイプは次のとおりです。

- *Person*
- *Organization*
- *Location*

関連性タイプ

RelationshipType	Entity1 のタイプ	Entity2 のタイプ	該当する関連性
<i>AffiliatedWith</i>	<i>Person</i>	<i>Organization</i>	<p><i>Person</i>エンティティと <i>Organization</i> エンティティの職業上または学術上の関連性を示します。</p> <p>次のような関連性またはこれらに近い関連性が成立します。</p> <ul style="list-style-type: none"> • <i>Person</i> 在学中または既卒者 <i>Organization</i> • <i>Person</i> 勤務中または過去に勤務 <i>Organization</i> • <i>Person</i> 発注者 <i>Organization</i> <p>注：これは、このタイプに該当する関連性の参考例です。</p> <p>例を次に示します。</p> <p>James has studied from the University of Toronto and works at ABC Corp.</p> <p>この場合、2つの関連性をパーシングできます。</p> <p>Entity1 = James、RelationshipType = <i>AffiliatedWith</i>、Entity2 = University of Toronto</p> <p>Entity1 = James、RelationshipType = <i>AffiliatedWith</i>、Entity2 = ABC Corp</p>

RelationshipType	Entity1 のタイプ	Entity2 のタイプ	該当する関連性
<i>LivesIn</i>	<i>Person</i>	<i>Location</i>	<p><i>Person</i> エンティティと <i>Location</i> エンティティの関連性を示します。</p> <p>次のような関連性が成立します。</p> <ul style="list-style-type: none"> • <i>Person</i> 滞在中または過去に滞在 <i>Location</i> • <i>Person</i> 転居済み <i>Location</i> • <i>Person</i> 誕生 <i>Location</i> • <i>Person</i> 死亡 <i>Location</i> <p>注：これは、このタイプに該当する関連性の参考例です。</p> <p>例を次に示します。</p> <p>John Jamison, a National Weather Service meteorologist in Galveston, reported that a massive hurricane was about to hit the East Coast the next day.</p> <p>Entity1 = John Jamison、RelationshipType = <i>LivesIn</i>、Entity2 = Galveston</p>
<i>OrgBasedIn</i>	<i>Organization</i>	<i>Location</i>	<p><i>Organization</i> のオフィスの少なくとも 1 つが <i>Location</i> にあることを示します。</p> <p><i>Location</i> は、支社、開発拠点、本社などのいずれかです。</p> <p>例を次に示します。</p> <p>HSBC Holdings Plc. is headquartered in London, United Kingdom.</p> <p>この場合、2 つの関連性をパーシングできます。</p> <p>Entity1 = HSBC Holdings Plc.、RelationshipType = <i>OrgBasedIn</i>、Entity2 = London</p> <p>Entity1 = HSBC Holdings Plc.、RelationshipType = <i>OrgBasedIn</i>、Entity2 = United States of America</p>

RelationshipType	Entity1 のタイプ	Entity2 のタイプ	該当する関連性
<i>LocatedIn</i>	<i>Location</i>	<i>Location</i>	<p>2つの場所の関連性 (地理的に一方が他方に含まれること) を示します。</p> <p>例 1 Canberra is the capital of Australia. この例では、以下が成立します。 Entity1 = Canberra、RelationshipType = <i>LocatedIn</i>、Entity2 = Australia</p> <p>例 2 India has as its capital New Delhi. この例では、以下が成立します。 Entity1 = India、RelationshipType = <i>LocatedIn</i>、Entity2 = New Delhi</p>
<i>Negative</i>	<i>Person</i> <i>Organization</i> <i>Location</i>	<i>Organization</i> <i>Location</i>	<p>上記の関連性タイプのいずれも2つのエンティティ間のパーシングで確認されなかったことを示します。</p> <p>例を次に示します。</p> <p>New Delhi and New York are good places to live in.</p> <p>この入力テキストをパーシングすると、識別されたエンティティをどう組み合わせても、サポートされている関連性タイプは見つかりません。したがって、次のように、識別されたエンティティ間には <i>Negative</i> 関連性タイプが成立すると判定されます。</p> <p>Entity1 = New Delhi、RelationshipType = <i>Negative</i>、Entity2 = New York</p>

注 : **Splitter** ステージを **Relationship Extractor** ステージの出力に接続して、識別された関連性タイプと、その関連性で結び付けられたエンティティのペアを抽出することができます。Splitter ステージは、このステージの階層出力をフラット出力に変換します。

例

複雑な入力テキストを使用すると、同じセンテンスについて複数の組み合わせで関連性タイプのペアがパーシングされることがあります。

例を次に示します。

James McCarthy has settled in New York, United States as director of ABC Technologies.

Relationship Extractor ステージで、ステージ オプションによって選択された関連性タイプを使用して入力テキストがパーシングされると、次の関連性が見つかります。

- 関連性 1 Entity1 = James McCarthy、Entity1 Type = *Person*、RelationshipType = *LivesIn*、Entity2 = New York、Entity2 Type = *Location*
- 関連性 2 Entity1 = James McCarthy、Entity1 Type = *Person*、RelationshipType = *AffiliatedWith*、Entity2 = ABC Technologies、Entity2 Type = *Organization*
- 関連性 3 Entity1 = ABC Technologies、Entity1 Type = *Organization*、RelationshipType = *OrgBasedIn*、Entity2 = United States、Entity2 Type = *Location*
- 関連性 4 Entity1 = ABC Technologies、Entity1 Type = *Organization*、RelationshipType = *OrgBasedIn*、Entity2 = New York、Entity2 Type = *Location*
- 関連性 5 Entity1 = James McCarthy、Entity1 Type = *Person*、RelationshipType = *LivesIn*、Entity2 = United States、Entity2 Type = *Location*
- 関連性 6 Entity1 = New York、Entity1 Type = *Location*、RelationshipType = *LocatedIn*、Entity2 = United States、Entity2 Type = *Location*

5 - ステージ リファレンス

このセクションの構成

Information Extraction コンポーネント	29
Read from Documents	29
Entity Extractor	34
Text Categorizer	37
Relationship Extractor	39

Information Extraction コンポーネント

Information Extraction モジュールには、次のステージが含まれています。

- **Read From Documents** — さまざまなファイル形式から構造化されていない入力データを読み取り、その内容を抽出します。
- **Entity Extractor** — 文字列として渡された構造化されていないデータから、名前や住所などのエンティティを抽出します。
- **Text Categorizer** — 構造化されていないコンテンツまたはプレーン テキスト (電子メール、ニュース記事、コメントなど) にカスタム カテゴリを割り当てます。そのカテゴリに関連する内容がどれだけ含まれているかに基づいて分類します。
- **Relationship Extractor** — エンティティ間の関連性を抽出します。

Read from Documents

Read from Documents は、さまざまなファイル形式から構造化されていない入力データを読み取り、内容を抽出するソースステージです。使用できるソースとして、法的文書、顧客からのフィードバック、製品レビュー、ニュース記事、ブログ、ソーシャル ネットワークなどがあります。**Read from Documents** は、作成者や作成日などのメタデータ フィールドの抽出も行います。抽出されたデータは、さまざまなタイプの処理 (特に、エンティティ抽出や文字列操作) に使用できます。また、構造化されていないテキストを検索するための検索インデックスの作成にも使用できます。

注：それぞれのドキュメントは、このステージ用の 1 つのレコードと見なされます。

入力

Read from Documents の入力は、単一のファイルまたはフォルダです。このステージは、以下のファイルの種類をサポートしています。

- テキスト
- PDF
- Microsoft Outlook
- Microsoft Word

- HTML

Read from Documents は、次の 3 つのタイプの抽出を実行します。

- ドキュメント — ドキュメント全体を使用
- ページ — ドキュメントの特定のページを使用
- 選択 — ドキュメントの選択部分を使用
- ブックマーク — PDF ドキュメントからのブックマークを使用

Read from Documents は、Information Extraction モジュールの一部です。

オプション

[ファイル プロパティ] タブ

以下の表に、Readfrom Documents が返す情報の種類を制御するオプションの一覧を示します。

表 1 : ReadfromDocuments のオプション

オプション	説明
サーバ名	使用する Spectrum Technology Platform サーバの名前を指定します。
ファイル/フォルダ名	ソース ドキュメントまたはフォルダのパスおよび名前です。フォルダを指す場合には、アスタリスクを ("**") ワイルドカード文字として使用して、フォルダ内のすべてのファイルを選択します。同一フォルダ内にある同じタイプの複数のファイルを指す場合は、ワイルドカード文字にファイル拡張子 ("*.pdf") を付け加えて使用します。
ファイル タイプ:	ソース ドキュメントのファイル タイプです。ソースを選択すると、自動的に選択されます。 <ul style="list-style-type: none"> • テキスト • PDF • Microsoft Outlook • Microsoft Word • HTML

オプション

説明

抽出タイプ	ドキュメント	ドキュメント全体を使用します。
	ページ	ドキュメントの特定のページを使用します。
	選択	ドキュメントの選択部分を使用します。
	ブックマーク	PDF ドキュメントからのブックマークを使用します。

ページの選択 「ページ」抽出タイプでのみ使用します。すべてのページまたはページ範囲を選択します。

選択された抽出 「選択」抽出タイプでのみ使用します。検索のタイプを指定します。

テキスト詳細 「選択」抽出タイプでのみ使用します。探し出すテキストを指定します。

開始テキストを除外する 「選択」抽出タイプと「開始テキスト」オプションの組み合わせでのみ使用します。入力された文字列を返されるデータから除外します。

終了テキストを指定する 「選択」抽出タイプでのみ使用します。探し出す終了テキストを指定します。

終了テキストを除外する 「選択」抽出タイプでのみ使用します。入力された文字列を返されるデータの末尾から除外します。

選択リターン 「選択」抽出タイプでのみ使用します。それぞれの結果で返す段落数を指定します。例えば、ここで"2"を選択した場合、それぞれの結果で返されるデータには、その結果の段落に後続の段落を加えた合計2つの段落が含まれることとなります。デフォルト値は1です。終了テキストが指定されると無効になります。

[フィールド] タブ

入力フィールドの定義を行うには、**[再生成]**をクリックします。

表 2 : 出力データ オプション

オプション	説明
属性名	入力フィールドに最も近い属性を示します。例えば、フィールドに日付情報が含まれていて、その情報を "Date" (日付) と呼んでいる場合は、そのフィールドに "Date" (日付) 属性が割り当てられます。この列は編集できません。
名前	フィールドの名前です。この列は編集できます。
タイプ	フィールドのデータ タイプ。
含める	どのフィールドを検索インデックスに含めるかを指定します。

出力

ReadfromDocuments ステージには、出力方向のポートが2つあります。1つのポートは、ステージによって読み取られたうえで入力された条件に基づいて返されたデータを取得します。このデータには、プレーンテキストまたはメタデータ (作成者、作成された日付など) を含めることができます。このポートは、入力方向のデータを読み取る任意のステージ (**Write to File**、**Write to XML** など) のほか、**Validate Address** や **Write to Search Index** のようなプライマリ ステージにも接続できます。また、ドキュメント内にある特定のエンティティ タイプに関する情報を返す場合には、**Information Extractor** ステージにも接続できます。「ドキュメント」抽出タイプを選択すると、出力にはフラット データが含まれます。「ページ」または「選択」の抽出タイプを選択すると、出力には階層データが含まれます。

もう 1つのポートは、データフローによって正しく処理されなかったレコードをすべて収集します。このポートをエラー ポートといいます。このポートからシンク ステージに送られるレコードは形式に誤りがあると考えられます。形式に誤りのあるレコードをキャプチャすることは、そのレコードに関する問題を特定するのに役立ちます。シンクをエラー ポートに接続すると、出力ファイルには、形式に誤りのあるレコードからのすべてのフィールドが含まれます。レコードの処理に失敗した理由を示す理由フィールドも含まれます。

表 3 : Unstructured Reader の出力

フィールド名	説明 / 有効な値
Author	通常、ドキュメントを作成または更新した人物の名前が含まれます。この情報は、ドキュメントのメタデータの一部です。
Bookmark	PDF 入力ファイルからのすべてのブックマークが含まれます。「ブックマーク」抽出タイプの場合のみ。
BookmarkNo	PDF 入力ファイルからのすべてのブックマークが含まれます。「ブックマーク」抽出タイプの場合のみ。
ContentLength	ドキュメントの長さを示します。この値は、選択される抽出タイプに応じて変化します。 ドキュメント ドキュメントのページ数です。 ページ 常に "1" であり、単一ページのコンテンツを表します。
Contents	抽出タイプによって異なります。例えば、「ドキュメント」抽出タイプの場合は、ドキュメント全体がフラット データとして出力されます。ページ、選択、ブックマークの各抽出タイプの場合は、階層データが出力されます。
ContentType	読み取られたドキュメントのタイプ (PDF、.txt など) を示します。
Creator	通常、ドキュメントを作成した人物の名前が含まれます。この情報は、ドキュメントのメタデータの一部です。
日付	ドキュメントが作成または最後に更新された日付を示します。
キーワード	ドキュメントのメタデータに指定されていたすべてのキーワードが含まれます。
言語	ドキュメントの記述言語を示します。

フィールド名	説明 / 有効な値
NPages	ドキュメントのページ数を示します。
PageContents	選択されたページの内容が含まれます。「ページ」抽出タイプの場合のみ。
PageNo	ブックマークのページ番号が含まれます。「ページ」抽出タイプの場合のみ。
Parent	XML ファイルの XPath に似た、ブックマークのパスが含まれます。「ブックマーク」抽出タイプの場合のみ。
ResourceName	ドキュメントのファイル名を示します。
SectionContents	選択されたセクションの内容が含まれます。「選択」抽出タイプの場合のみ。
SectionNo	そのセクションのドキュメント内での番号を示します。「選択」抽出タイプの場合のみ。
Subject	ドキュメントのメタデータに指定されていたドキュメントの主題が含まれます。
タイトル	ドキュメントのメタデータに指定されていたドキュメントのタイトルが含まれます。

Entity Extractor

Entity Extractorは、構造化されていないデータの文字列 (プレーン テキスト) から、名前や住所などのエンティティを抽出します。

入力のタイプによって精度が異なるため、選択されたタイプに対して一部のエンティティは返されない可能性があります。Entity Extractor は自然言語処理を使用するため、ニュース記事やプロ

グからの文法的に正しい文を含む文字列は、名前や日付の単純なリストよりも、高い精度で名前が返される可能性があります。

入力

Entity Extractor は、構造化されていない文字列データを入力として受け取ります。また、構造化されていないドキュメントからエンティティを抽出する場合は、**Read from Documents** ステージを入力として使用できます。**Read from Documents** ステージは、ドキュメントを読み取り、ユーザ定義の設定に基づいてテキストを返します。**Entity Extractor** は、選択されたエンティティに基づいてこのテキストから必要な情報を抽出します。

表 4: 入力フォーマット

フィールド名	説明
PlainText	情報の抽出元となるデータを表す構造化されていない文字列です。

オプション

Entity Extractor オプションを使用すると、入力文字列から情報を抽出するために使用するエンティティを選択することができます。デフォルトで、情報の抽出に使用されるエンティティタイプは **Person** と **Address** です。ただし、**[クイック追加]** 機能を使用して次の 15 の設定済みエンティティから任意のものまたはすべてを選択できます。

オプション名	説明
次の値でシステム デフォルト オプションを上書きします。	<p>チェック ボックスを選択して、デフォルトのエンティティタイプ Address と Person を上書きします。</p> <p>チェック ボックスを選択すると、[クイック追加] ボタンが有効になります。このボタンをクリックして、テキストの抽出に必要なエンティティを選択します。</p> <p>選択したエンティティは、[エンティティタイプ] リストに追加されます。</p>

オプション名	説明
エンティティ タイプ	<p>構造化されていない文字列から抽出するデータのタイプを指定します。</p> <p>Address</p> <p>CreditCard</p> <p>Date</p> <p>Email</p> <p>HashTag</p> <p>ISBN</p> <p>Location</p> <p>Mention</p> <p>Organization</p> <p>Person</p> <p>Phone</p> <p>ProperNouns</p> <p>SSN</p> <p>WebAddress</p> <p>ZipCode</p>
出力エンティティ数	<p>特定のエンティティが出力に現れる回数を返すかどうかを指定します。</p> <p>true 構造化されていない文字列内に見つかったエンティティの数を返します。</p> <p>false 構造化されていない文字列内に見つかったエンティティの数を返しません。</p>

出力

Entity Extractor からの出力は、入力文字列内に見つかったエンティティの一覧です。例えば、"Person" というエンティティ タイプを選択した場合、出力は入力文字列内に見つかった名前の一覧になります。同様に、**[エンティティ タイプ]**として "Date" を選択した場合は、入力文字列内に見つかった日付の一覧が出力になります。

それぞれのエンティティ (名前、住所、日付など) は、たとえ入力文字列内に複数回現れていても、一度しか返されません。

入力文字列に特定のエンティティが現れた回数を確認するには、**[Entity Extractor オプション]** ウィンドウで **[出力エンティティ数]** オプションを選択します。

フィールド名	説明
Text	文字列から抽出されたテキストです。
Type	抽出されたテキストのエンティティ タイプです。次のいずれかです。 Address CreditCard Date Email HashTag ISBN Location Mention Organization Person Phone ProperNouns SSN WebAddress ZipCode
Count	回数を返すオプションが有効になっている場合は、このフィールドに、その特定のエンティティが入力に現れた回数が含まれます。例えば、Nameエンティティを返すことを選択していて、入力テキストに "John" という名前のインスタンスが 5 つ含まれている場合、出力には "John" が 1 回だけ含まれ、エンティティ タイプは "Name" に、出力回数は "5" になります。

Text Categorizer

このステージでは、構造化されていないコンテンツまたはプレーンテキスト (電子メール、ニュース記事、コメントなど) に、そのコンテンツのマッチングレベルに基づいて、カスタム カテゴリ

を割り当てます。定義されているカテゴリの一覧が表示されるので、分類に必要なカテゴリをそこから選択できます。ただし、カテゴリライザ モデルをデータに合わせてトレーニングすることにより、これらのカテゴリを作成する必要があります。詳細については、[テキスト分類の概要](#)（6ページ）を参照してください。

入力

このステージは、構造化されていない文字列データを入力として受け取ります。また、構造化されていないドキュメントからテキストを分類する場合は、**Read from Documents** ステージを入力として使用できます。**Read from Documents** ステージは、ドキュメントを読み取り、ユーザー定義の設定に基づいてテキストを返します。これが **Text Categorizer** ステージに読み込まれ、必要な出力が提供されます。

表 5: 入力フォーマット

フィールド名	説明
PlainText	情報の抽出元となるデータを表す構造化されていない文字列です。

オプション

Text Categorizer オプションを使用すると、入力文字列データの分類基準として使用するパラメータを選択することができます。分類のモデルと、出力に含めるマッチ レベルの数を選択できます。例えば、最も近い一致のみ、最も近いものと 2 番目に近いもの、などを選択できます。

オプション名	説明
次の値でシステム デフォルト オプションを上書きします。	デフォルト オプションを上書きし、カテゴリライザを [カテゴリライザ名] ドロップダウンから選択します。
カテゴリライザ名	テキスト分類に使用するモデルを指定します。テキスト分類フェーズでトレーニングしたすべてのモデルが一覧に含まれます。 注：詳細については、 モデルのトレーニング （11ページ）を参照してください。

オプション名	説明
カテゴリ数	出力に含めるカテゴリのマッチレベル数。例えば、1を選択すると最も近い一致のみが表示され、2を選択すると最も近い一致と2番目に近い一致が表示されず。 注: 最大値は、モデルのトレーニング中に指定した異なるクラスの数に対応します。

出力

出力は、入力文字列の内容を分類したカテゴリと、そのランクを一覧で示します。このランクは、入力内容のカテゴリに対するマッチングのレベルを表します。例えば、1はカテゴリに最も近い一致を意味し、2は最も近い一致と2番目に近いものを意味します。

フィールド名	説明
Category	入力ファイルに含まれる各レコードの予測カテゴリ。
Rank	最高スコアから最低スコアまでのカテゴリのランク。

Relationship Extractor

Relationship Extractor ステージを使用すると、ソース コンテンツ内で識別されたエンティティの関連性タイプを識別できます。

Relationship Extractor ステージは、以下を識別します。

1. Entity1
2. Entity1 のタイプ
3. 関連性タイプ
4. Entity2

5. Entity2 のタイプ

重要： このステージは、可能な限り高い精度を達成しようとしながら、入力テキストに含まれる2つのエンティティ間に成立する関連性タイプを識別します。ただし、2つのエンティティ間の正確な関連性ではない関連性も、入力テキストの複雑なセンテンスをパーシングする過程で識別されます。

入力

Relationship Extractor ステージは、自然言語の文字列データを入力として受け取り、エンティティを識別し、さらにエンティティの各ペア間に成立する関連性タイプを識別します。

入力テキストが構造化されていないドキュメントから取得される場合は、**Read from Documents** ステージをソース ステージとして使用します。**Read from Documents** ステージは、ドキュメントを読み取り、ユーザ定義の設定に基づいてテキストを返します。

Relationship Extractor ステージは、すべてのエンティティを識別し、さらにエンティティの各ペア間に成立する関連性タイプを識別します。

表 6 : 入力フォーマット

フィールド名	説明
PlainText	エンティティの各ペアに成立する関連性タイプの識別に <input data-bbox="164 1108 1421 1333" type="text"/> として使用する、構造化されていない文字列データ。

オプション

Relationship Extractor ステージのオプションを使用すると、入力テキストから識別する関連性タイプを指定することができます。

デフォルトで、次の関連性タイプが識別されます。

1. *AffiliatedWith*
2. *LivesIn*
3. *OrgBasedIn*
4. *LocatedIn*

オプション名	説明
--------	----

次の値でシステム デフォルト オプションを上書きします。	<p>識別されたデフォルト関連性タイプを上書きするチェック ボックスを選択し、入力テキストから識別して抽出する関連性タイプを指定します。</p> <p>チェック ボックスを選択すると、[クイック追加] ボタンが有効になります。[クイック追加] をクリックして、テキストから識別する関連性タイプを選択します。</p> <p>選択したエンティティは、[関連性タイプ] リストに追加されます。</p>
------------------------------	--

出力

Relationship Extractor から出力される情報は、入力文字列に見つかったエンティティのペア間に識別された一連の関連性です。

例えば、ステージのオプションで、抽出する関連性タイプとして *LivesIn* と *OrgBasedIn* を選択した場合、入力テキストから識別された *Person LivesIn Location* と *Organization OrgBasedIn Location* の組み合わせがすべて出力されます。

エンティティのペアに識別された各関連性タイプは一度のみ出力されます。

抽出されたエンティティと関連性の各セットには、次の情報が含まれます。

フィールド名	説明
--------	----

Entity1	入力テキストから抽出されたエンティティ ペアの 1 つ目のエンティティ。
Entity1 Type	<p>入力テキストから抽出されたエンティティ ペアの 1 つ目のエンティティのエンティティ タイプ。</p> <p>エンティティ タイプは、次のいずれかです。</p> <ul style="list-style-type: none"> • <i>Person</i> • <i>Organization</i> • <i>Location</i>

フィールド名	説明
Type	<p>Entity1 と Entity2 の間に識別された関連性タイプ。</p> <p>関連性タイプの詳細については、関連性タイプ (24ページ) を参照してください。</p> <p>注：抽出する関連性タイプとしてステージのオプションによって選択したタイプのみが識別され、表示されます。</p>
Entity2	<p>入力テキストから抽出されたエンティティ ペアの 2 つ目のエンティティのエンティティ タイプ。</p>
Entity2 Type	<p>入力テキストから抽出されたエンティティ ペアの 2 つ目のエンティティのエンティティ タイプ。</p> <p>エンティティ タイプは、次のいずれかです。</p> <ul style="list-style-type: none">• <i>Person</i>• <i>Organization</i>• <i>Location</i>

著作権に関する通知

© 2017 Pitney Bowes Software Inc. All rights reserved. MapInfo および Group 1 Software は Pitney Bowes Software Inc. の商標です。その他のマークおよび商標はすべて、それぞれの所有者の資産です。

USPS® 情報

Pitney Bowes Inc. は、ZIP + 4® データベースを光学および磁気媒体に発行および販売する非独占的ライセンスを所有しています。CASS、CASS 認定、DPV、eLOT、FASTforward、First-Class Mail、Intelligent Mail、LACS^{Link}、NCOA^{Link}、PAVE、PLANET Code、Postal Service、POSTNET、Post Office、RDI、Suite^{Link}、United States Postal Service、Standard Mail、United States Post Office、USPS、ZIP Code、および ZIP + 4 の各商標は United States Postal Service が所有します。United States Postal Service に帰属する商標はこれに限りません。

Pitney Bowes Inc. は、NCOA^{Link}® 処理に対する USPS® の非独占的ライセンスを所有しています。

Pitney Bowes Software の製品、オプション、およびサービスの価格は、USPS® または米国政府によって規定、制御、または承認されるものではありません。RDI™ データを利用して郵便送料を判定する場合に、使用する郵便配送業者の選定に関するビジネス上の意思決定が USPS® または米国政府によって行われることはありません。

データ プロバイダおよび関連情報

このメディアに含まれて、Pitney Bowes Software アプリケーション内で使用されるデータ製品は、各種商標によって、および次の 1 つ以上の著作権によって保護されています。

© Copyright United States Postal Service. All rights reserved.

© 2014 TomTom. All rights reserved. TomTom および TomTom ロゴは TomTom N.V の登録商標です。

© 2016 HERE

Fuente: INEGI (Instituto Nacional de Estadística y Geografía)

電子データに基づいています。© National Land Survey Sweden.

© Copyright United States Census Bureau

© Copyright Nova Marketing Group, Inc.

このプログラムの一部は著作権で保護されています。© Copyright 1993-2007 by Nova Marketing Group Inc. All Rights Reserved

© Copyright Second Decimal, LLC

© Copyright Canada Post Corporation

この CD-ROM には、Canada Post Corporation が著作権を所有している編集物からのデータが収録されています。

© 2007 Claritas, Inc.

Geocode Address World データ セットには、
<http://creativecommons.org/licenses/by/3.0/legalcode> に存在するクリエイティブ コモンズ アトリビューション ライセンス (「アトリビューション ライセンス」) の下に提供されている GeoNames Project (www.geonames.org) からライセンス供与されたデータが含まれています。お客様による GeoNames データ (Spectrum™ Technology Platform ユーザ マニュアルに記載) の使用は、アトリビューション ライセンスの条件に従う必要があります。お客様と Pitney Bowes Software, Inc. との契約と、アトリビューション ライセンスの間に矛盾が生じる場合は、アトリビューション ライセンスのみに基づいてそれを解決する必要があります。お客様による GeoNames データの使用に関しては、アトリビューション ライセンスが適用されるためです。



3001 Summer Street
Stamford CT 06926-0700
USA

www.pitneybowes.com