

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6511221号

(P6511221)

(45) 発行日 令和1年5月15日 (2019.5.15)

(24) 登録日 平成31年4月12日 (2019.4.12)

(51) Int. Cl.

F 1

G 0 6 F 17/28 (2006.01)

G 0 6 F 17/28 6 7 2

請求項の数 9 (全 22 頁)

|               |                               |           |                     |
|---------------|-------------------------------|-----------|---------------------|
| (21) 出願番号     | 特願2013-526173 (P2013-526173)  | (73) 特許権者 | 502208397           |
| (86) (22) 出願日 | 平成23年8月26日 (2011.8.26)        |           | グーグル エルエルシー         |
| (65) 公表番号     | 特表2013-540304 (P2013-540304A) |           | アメリカ合衆国 カリフォルニア州 94 |
| (43) 公表日      | 平成25年10月31日 (2013.10.31)      |           | 043 マウンテン ビュー アンフィシ |
| (86) 国際出願番号   | PCT/US2011/049352             |           | アター パークウェイ 1600     |
| (87) 国際公開番号   | W02012/027672                 | (74) 代理人  | 100108453           |
| (87) 国際公開日    | 平成24年3月1日 (2012.3.1)          |           | 弁理士 村山 靖彦           |
| 審査請求日         | 平成26年8月19日 (2014.8.19)        | (74) 代理人  | 100110364           |
| 審査番号          | 不服2017-4592 (P2017-4592/J1)   |           | 弁理士 実広 信哉           |
| 審査請求日         | 平成29年4月3日 (2017.4.3)          | (74) 代理人  | 100133400           |
| (31) 優先権主張番号  | 61/377,440                    |           | 弁理士 阿部 達彦           |
| (32) 優先日      | 平成22年8月26日 (2010.8.26)        | (72) 発明者  | サッシャ・ペー・ブラヴェル       |
| (33) 優先権主張国   | 米国 (US)                       |           | スイス・CH-3012・ベルン・ランガ |
|               |                               |           | スシュトラーセ・72          |

最終頁に続く

(54) 【発明の名称】 入力テキスト文字列の変換

(57) 【特許請求の範囲】

【請求項1】

第1の形式の、複数の語を有する入力文字列を受信するステップであって、前記入力文字列が地理的名称を含み、前記入力文字列は、前記地理的名称に対応する物理的世界における実体の特徴を各々が示す複数の地理的特徴タイプから選択された地理的特徴タイプを伴う、ステップと、

前記入力文字列を前記第1の形式から第2の形式に変換するステップ、であって、

前記入力文字列が前記複数の地理的特徴タイプのうちの第1の地理的特徴タイプを伴う場合に、前記第1の地理的特徴タイプを伴う前記入力文字列内の1つまたは複数の語を前記第2の形式の1つまたは複数の翻訳された語に翻訳し、前記入力文字列が前記第1の形式にとどまる語を更に含む場合にはこれを前記第2の形式に書き直す、ステップと、

前記入力文字列が前記複数の地理的特徴タイプのうちの前記第1の地理的特徴タイプとは異なる第2の地理的特徴タイプを伴う場合に、前記複数の地理的特徴タイプのうちの前記第2の地理的特徴タイプを伴う1つまたは複数の語を前記第2の形式の1つまたは複数の語に書き直す、ステップと

を含むステップと、

少なくとも前記翻訳された語と前記書き直された語とを結び付けて、前記第2の形式の混合型出力文字列を形成する、ステップと

を含む、

前記翻訳することは、前記第1の形式の語を、前記第1の形式の語と同じ意味を有する第

10

20

2の形式の語に変換することであり、

前記書き直すことは、前記第1の形式の語を、前記第1の形式の語と同じ発音を有する第2の形式の語に変換することである、データ処理装置によって実行される方法。

【請求項2】

前記第1の形式および前記第2の形式は、それぞれ第1の書記体系および第2の書記体系である、請求項1に記載の方法。

【請求項3】

前記第1の形式および前記第2の形式は、それぞれ第1の自然言語および第2の自然言語である、請求項1に記載の方法。

【請求項4】

前記書き直すことは、  
前記文字列を複数のトークンにトークン化するステップと、  
各トークンを前記第1の形式から第2の形式に書き直すステップと、  
前記第2の形式の前記書き直されたトークンを結び付けて、前記第2の形式の書き直された出力文字列を形成するステップと  
を含む、請求項1に記載の方法。

【請求項5】

第1の形式の、複数の語を有する入力文字列を受信する動作であって、前記入力文字列が地理的名称を含み、前記入力文字列は、前記地理的名称に対応する物理的世界における実体の特徴を各々が示す複数の地理的特徴タイプから選択された地理的特徴タイプを伴う、動作と、

前記入力文字列を前記第1の形式から第2の形式に変換する、動作であって、

前記入力文字列が前記複数の地理的特徴タイプのうちの第1の地理的特徴タイプを伴う場合に、前記第1の地理的特徴タイプを伴う前記入力文字列内の1つまたは複数の語を、前記第2の形式の1つまたは複数の翻訳された語に翻訳し、前記入力文字列が前記第1の形式にとどまる語を更に含む場合にはこれを前記第2の形式に書き直す、動作と、

前記入力文字列が前記複数の地理的特徴タイプのうちの前記第1の地理的特徴タイプとは異なる第2の地理的特徴タイプを伴う場合に、前記複数の地理的特徴タイプのうちの前記第2の地理的特徴タイプを伴う1つまたは複数の語が前記第2の形式の1つまたは複数の語に書き直される、動作と

を含む動作と、

少なくとも前記翻訳された語と前記書き直された語とを結び付けて、前記第2の形式の混合型出力文字列を形成する、動作と

を含む動作を実行するたに對話するように動作可能な1つまたは複数のコンピュータを含み、

前記翻訳することは、前記第1の形式の語を、前記第1の形式の語と同じ意味を有する第2の形式の語に変換することであり、

前記書き直すことは、前記第1の形式の語を、前記第1の形式の語と同じ発音を有する第2の形式の語に変換することである、システム。

【請求項6】

前記第1の形式および前記第2の形式は、それぞれ第1の書記体系および第2の書記体系である、請求項5に記載のシステム。

【請求項7】

前記第1の形式および前記第2の形式は、それぞれ第1の自然言語および第2の自然言語である、請求項5に記載のシステム。

【請求項8】

前記書き直すことは、

前記文字列を複数のトークンにトークン化する動作と、

各トークンを前記第1の形式から第2の形式に書き直す動作と、

前記第2の形式の前記書き直されたトークンを結び付けて、前記第2の形式の書き直され

10

20

30

40

50

た出力文字列を形成する動作と  
を含む、請求項5に記載のシステム。

【請求項9】

データ処理装置によって実行されるときに、前記データ処理装置に対し、

第1の形式の、複数の語を有する入力文字列を受信する動作であって、前記入力文字列が地理的名称を含み、前記入力文字列は、前記地理的名称に対応する物理的世界における実体の特徴を各々が示す複数の地理的特徴タイプから選択された地理的特徴タイプを伴う、動作と、

前記入力文字列を前記第1の形式から第2の形式に変換する、動作であって、

前記入力文字列が前記複数の地理的特徴タイプのうちの第1の地理的特徴タイプを伴う場合に、前記第1の地理的特徴タイプを伴う前記入力文字列内の1つまたは複数の語を、前記第2の形式の1つまたは複数の翻訳された語に翻訳し、前記入力文字列が前記第1の形式にとどまる語を更に含む場合にはこれを前記第2の形式に書き直す、動作と、

前記入力文字列が前記複数の地理的特徴タイプのうちの前記第1の地理的特徴タイプとは異なる第2の地理的特徴タイプを伴う場合に、前記複数の地理的特徴タイプのうちの前記第2の地理的特徴タイプを伴う1つまたは複数の語が前記第2の形式の1つまたは複数の語に書き直される、動作と

を含む動作と、

少なくとも前記翻訳された語と前記書き直された語とを結び付けて、前記第2の形式の混合型出力文字列を形成する、動作と

を含む動作を実行させる命令を含み、

前記翻訳することは、前記第1の形式の語を、前記第1の形式の語と同じ意味を有する第2の形式の語に変換することであり、

前記書き直すことは、前記第1の形式の語を、前記第1の形式の語と同じ発音を有する第2の形式の語に変換することである、コンピュータプログラムで符号化されたコンピュータ記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本明細書は、入力文字列の変換に関する。

【背景技術】

【0002】

従来型のテキストは特定の形式、たとえば、特定の書記体系および特定の自然言語と結び付いている。1つの書記体系は、1つまたは複数の言語を表すために使用されうる。たとえば、(ローマ字を使用して表される)ラテン語の書記体系は、自然言語の英語のほか、たとえばローマ字化された中国語(たとえば、ピン音)で使用されるように、自然言語の中国語に使用されうる。同様に、複数の書記体系が、同じ言語を表すために使用されうる。たとえば、漢字およびピン音の書記体系が、中国語を表すために使用されうる。入力文字列を形式間で(たとえば、ある書記体系から別の書記体系に、またはある自然言語から別の自然言語に)変換することは、入力文字列の翻訳または書き直しを伴うことがある。

【発明の概要】

【課題を解決するための手段】

【0003】

本明細書は、第1の形式から第2の形式へのテキスト入力文字列の変換に関係する技術について説明する。

【0004】

テキストおよび音声の国際化および翻訳において、いくつかの単語が、それらの意味に基づいて翻訳される(たとえば、英語の「high」は日本語の「高い」に、英語の「bridge」は「橋」に翻訳されうる)。他の単語は、それらの音または正字法、たとえば、ある言語を書くにあたり特定の書記体系を使用するルールに基づいて書き直される。たとえば、

英語名「Highbridge」は、日本語「ハイブリッジ」に書き直される。「高橋」という翻訳された形式にはならない。混合形式は、意味的に入力文字列の一部分を翻訳し、入力文字列の他の部分を書き直すことを含む。たとえば、「Highbridge Park」は、日本語では「ハイブリッジ公園」という混合形式になることがあり、この場合、「ハイブリッジ」が書き直し部分であり、「公園」が翻訳部分である。

**【0005】**

本明細書は、語の入力文字列(an input string of terms)を第1の形式から第2の形式に、たとえば、ある自然言語または書記体系から別の自然言語または書記体系に変換するための技法について説明する。変換は、たとえば、書き直しおよび混合型翻訳技法を使用して実行できる。語の入力文字列を第1の形式から第2の形式に変換することは、第1の言語の書記体系から第2の言語の書記体系に変換することのほか、同じ言語において2つの書記体系の間で変換することを含みうる。

10

**【0006】**

いくつかの実装形態では、複数の語を含む入力文字列の場合、機械変換を実行して、ある形式による入力文字列を別の形式の出力文字列に変換する。機械変換は、入力文字列における語の書き直しおよび翻訳の混合である出力文字列を生成することを含みうる。たとえば、語の入力文字列は、1つまたは複数の語が翻訳されるべきか否かを判断するためのルールを使用することと、他の語に対して書き直しを実行するための言語固有のルールを使用することを含みうる。

**【0007】**

20

一般に、本明細書で説明する対象の一態様は、第1の形式の、複数の語を有する入力文字列を受信するアクションと、入力文字列を第1の形式から第2の形式に変換するアクションであって、1つまたは複数のルールを入力文字列に適用して、複数の語よりも少ない翻訳用の1つまたは複数の語を識別するアクションと、識別された1つまたは複数の語を、第2の形式の1つまたは複数の翻訳された語に翻訳するアクションと、複数の語の残りの語を、第2の形式の書き直された語に書き直すアクションとを含むアクションと、翻訳された語と書き直された語とを結び付けて、第2の形式の混合型出力文字列を形成するアクションとを含む方法に具現化されうる。本態様の他の実施形態は、各々が本方法のアクションを実行するように構成された、対応するコンピュータシステム、装置、および1つまたは複数のコンピュータ記憶デバイスに記録されたコンピュータプログラムを含む。1つまたは複数のコンピュータのシステムは、動作中にシステムに対しアクションを実行させる、システムにインストールされているソフトウェア、ファームウェア、ハードウェア、またはこれらの組合せを有することによって、特定の動作またはアクションを実行するように構成されうる。1つまたは複数のコンピュータプログラムは、データ処理装置によって実行されるときに、かかる装置に対しアクションを実行させる命令を含むことによって、特定の動作またはアクションを実行するように構成されうる。

30

**【0008】**

これらおよび他の実施形態は、以下の特徴のうちの1つまたは複数の随意に含むことができる。第1の形式および第2の形式は、それぞれ第1の書記体系および第2の書記体系である。第1の形式および第2の形式は、それぞれ第1の自然言語および第2の自然言語である。入力文字列は、タイプを含み、1つまたは複数のルールを入力文字列に適用することは、入力文字列のタイプと合致するタイプを有する1つまたは複数のルールを識別することと、入力文字列のタイプと合致するルールにおいて、入力文字列が、それぞれの1つまたは複数の合致するルールの文字列パターンと合致するか否かを判断することを含む。各ルールは、それぞれの出力形式において複数のそれぞれのルール出力を含む。合致するルールは、第1の形式から第2の形式に翻訳されるルールパターンの1つまたは複数の語を含む。残りの語を書き直すことは、文字列を複数のトークンにトークン化することと、各トークンを第1の形式から第2の形式に書き直すことと、第2の形式の書き直されたトークンを結び付けて、第2の形式の書き直された出力文字列を形成することを含む。

40

**【0009】**

50

一般に、本明細書で説明する対象の一態様は、第1の形式の、書き直される複数の語を有する文字列を受信するアクションと、文字列を複数のトークンにトークン化するアクションと、各トークンを第1の形式から第2の形式に書き直すアクションと、1つまたは複数の形式固有のルールを、第2の形式の書き直されたトークンに適用するアクションと、第2の形式の書き直されたトークンを結び付けて、第2の形式の書き直された出力文字列を形成するアクションとを含む方法に具現化されうる。本態様の他の実施形態は、各々が本方法のアクションを実行するように構成された、対応するコンピュータシステム、装置、および1つまたは複数のコンピュータ記憶デバイスに記録されたコンピュータプログラムを含む。1つまたは複数のコンピュータのシステムは、動作中にシステムに対しアクションを実行させる、システムにインストールされているソフトウェア、ファームウェア、ハードウェア、またはこれらの組合せを有することによって、特定の動作またはアクションを実行するように構成されうる。1つまたは複数のコンピュータプログラムは、データ処理装置によって実行されるときに、かかる装置に対しアクションを実行させる命令を含むことによって、特定の動作またはアクションを実行するように構成されうる。

10

#### 【0010】

これらおよび他の実施形態は、以下の特徴のうちの1つまたは複数を随意に含むことができる。文字列をトークン化することは、文字列を単語トークンに分割することを含む。第1の形式は第1の書記体系であり、第2の形式は第2の書記体系である。第1の形式は第1の自然言語であり、第2の形式は第2の自然言語である。形式固有のルールは、形態素または単語の境界で生じる様々な音韻過程に関係する。結び付けは、出力形式および1つまたは複数の言語ルールに基づいて、出力語の1つまたは複数の対の間に追加の文字を加えることを含む。各トークンを書き直すことは、1つまたは複数の有限状態トランスデューサを使用して、第1の形式および第2の形式の発音表記を生成することを含む。

20

#### 【0011】

本明細書で説明する対象の特定の実施形態を実施して、以下の利点のうちの1つまたは複数を実現できる。変換の精度は、書き直しのみ、または翻訳のみの場合と比較して、書き直しおよび意味的翻訳の混合型を使用することで向上する。書き直しは、個々の語を別々に翻訳する場合と比較して、他の語の文脈を使用することで改善する。名前が書き写されている実体に関する情報を組み込むことで、書き直しの精度が向上する。たとえば、「Menlo Park」が公園を指すのか、それとも都市を指すのかを知っていることは、変換される出力に影響しうる。

30

#### 【0012】

本明細書で説明する対象の1つまたは複数の実施形態の詳細を、添付の図面および以下の説明で示す。対象の他の特徴、態様、および利点は、説明、図面、および特許請求の範囲から明らかになる。

#### 【図面の簡単な説明】

#### 【0013】

【図1】入力文字列を変換するための例示的な方法の流れ図である。

【図2】入力文字列の機械変換のための例示的な方法の流れ図である。

【図3】入力文字列の語を書き直すための例示的な方法の流れ図である。

40

【図4】例示的なシステムアーキテクチャである。

#### 【発明を実施するための形態】

#### 【0014】

様々な図面における同じ参照番号および表示は、同じ要素を示す。

#### 【0015】

図1は、入力文字列を変換するための例示的な方法100の流れ図である。便宜上、方法100は、方法100を実行する、1つまたは複数のコンピューティングデバイスを含むシステムに関して説明する。特に、方法100は(たとえば、マップビューにおける)提示用の地理的データを処理することに関する動作について説明するが、同様のアクションを他のタイプのデータに対して実行することもできる。

50

## 【0016】

システムは、第1の書記体系における1つまたは複数の入力文字列を受信する(102)。入力文字列は、第1の形式から第2の形式に(たとえば、第1の書記体系から第2の書記体系に)変換される。第1の形式および第2の形式は、同じ言語または異なる言語を表しうる。いくつかの実装形態では、1つまたは複数の入力文字列が、地理的特徴の集合から(たとえば、地理的特徴のデータベースから)受信される。これらの地理的特徴は、たとえば、政治的指定(political designation)(たとえば、都市または州の名前)、観光目的地、または公園を含みうる。

## 【0017】

各地理的特徴は、物理的世界における対応する実体に関する情報(たとえば、特徴タイプ)により注釈されるか、明示されることがある。情報はタイプのカテゴリ、たとえば、「都市」、「公園」、または「観光名所」を含むことができる。いくつかの実装形態では、タイプ分けは、継承のないタイプの単純な一覧である。他の実装形態では、タイプ分けは単一の継承ツリー階層を使用した手持ち式(handled)であり、各タイプは別のタイプのサブタイプであり、その他のタイプのサブタイプではない1つまたは複数の根タイプを例外とする。さらにいくつかの他の実装形態では、タイプ分けは複数の継承ツリーを使用する手持ち式であり、1つのタイプは他のタイプのサブタイプでないことも、1つまたは複数の他のタイプのサブタイプであることもある。以下で図2に関してさらに詳しく説明するように、ルールを生成するために特徴が使用されうる。

## 【0018】

加えて、タイプ階層は、特徴タイプについて説明する自由形式テキストの特徴ごとの複数のタイプタグを許容しうる。タイプ分けのほかに、特徴は、示された物理的実体に関する他の情報、たとえば、都市の人口、または建物の高さにより明示されることもある。たとえば、一定の規模の都市のみに、または大きい建物の名前だけにルールを適用するとき、生成される書き直しに影響を与えるために、この他の非タイプ分け情報が使用されることもある。

## 【0019】

地理的特徴の例が使用される一方で、他のデータ、たとえば、ビジネスレジストリまたは特定のタイプに関する外部情報が知られているか、導出されうる他のデータ(たとえば、製品名、個人)も同様に処理されうる。たとえば、企業名は適用可能な様々な企業特徴と結び付きうる、(たとえば、「Inc」を付けて株式会社(corporation)であることを表す)。

## 【0020】

システムは、随意に事前処理を実行する(104)。事前処理は、入力文字列に対して実行される複数の動作を含みうる。事前処理は、入力文字列に対して直接実行されるアクション、または後で入力文字列に適用するためのデータベースもしくは他の情報の集合(たとえば、辞書)を生成することを伴うこともある。

## 【0021】

いくつかの実装形態では、いくつかの処理ステップは、提示用の地理的データを処理するためのより大きいパイプラインの一部である。たとえば、地理的データの場合、事前処理は、重複する特徴を統合および/または除去すること、近隣データセットからの道を繋ぎ合わせる(stitching)こと、既存の特徴の形から新しい特徴を合成すること、形状の不規則(たとえば、単一点の位置または不整合な形状を有する通りなどの提供されたデータにおける誤り)を改めること、または特徴の属性を取り入れることを含みうる。

## 【0022】

近隣データセットからの道を繋ぎ合わせることは、異なるデータセットが使用されうる異なる地域間で地理的データを訂正することを指し、たとえば、第1の国と第2の国との間の境界を越える道は、第1の国のデータセットと第2の国のデータセットとの間で整合していないことがある。この事前動作は、総じて提示用の地理的データを処理するためのパイプラインの一部でありうる。地理的データの特徴を合成することは、たとえば、国および

地方から英語圏の境界を合成すること、たとえば、米国+オーストラリア+ニュージーランド+カナダ-ケベック州、を含む。これは、どの地域が入力文字列の変換を必要としているかを定義し、国または他の地政学的境界以外の情報に基づかない可能性のある部分を定義するのに役立つ。

【0023】

事前処理段階の一部として説明したが、これらのタスクは、実行されるタスクのタイプに応じて、後のプロセスの一部として(たとえば、書き直し中に)、または別個の動作として、複合の手順(complex sequence)で実行されうる。たとえば、図3に関して後述する書き直し動作中に、人間が入力した翻訳が適用されることがある。あるいは、いくつかの実装形態では、事前処理が実行されない(たとえば、特徴をマッピングする地理的動作は、別個に処理可能であり、または非地理的データセットには不要である)。

10

【0024】

システムは、入力文字列について人間が入力した翻訳を識別する(106)。これらは、不規則な翻訳を有する周知の語についての人間によって識別される翻訳である。たとえば、フランス語名が

【0025】

【数1】

"Genève"

【0026】

であるスイスの都市の場合、このステップは、英語名「Geneva」、ドイツ語「Genf」、イタリア語「Ginevra」、スロバキア語

20

【0027】

【数2】

"Ženeva"

【0028】

、ロシア語

【0029】

【数3】

30

"Женева"

【0030】

などを加えることがある。同様に、英語名が「California」である米国の州の場合、このステップは、ドイツ語名「Kalifornien」を加えることがある。これにより、対応する語を追加することで入力データを修正することができ、スペルミスまたは入力テキストを処理するのに使用される1つまたは複数のルールによって識別されていない語の言語版を使用することによる悪い変換の可能性が低下する。たとえば、1つの特定の通りの名前が、たとえばデータ提供者の問題により、誤字を有したり、誤って称されたりすることがある。これらの識別された翻訳は、たとえば、図2～図3に関して後述する書き直しまたは翻訳の動作を実行する際に使用するために、データベースまたは辞書に入力される。

40

【0031】

システムは、1つまたは複数の語について辞書の翻訳を識別する(108)。辞書の翻訳は、同じ名前を有する入力テキストの集合内のすべての語に適用される。たとえば、米国には「City Hall」と呼ばれる建物が沢山ある。辞書の表現は、その名前を調べ、ドイツ語の「Rathaus」、フランス語の

【0032】

【数4】

"Hôtel de Ville"

50

## 【0033】

、日本語の「役所」などの名前を、合致する名前を有するすべての語に入れることがある。こうして、特定の特徴または語に関する複数の翻訳が、文字列を変換する際に使用するために識別される。

## 【0034】

システムは、各入力文字列に対して機械変換を実行する(110)。機械変換は、第1の形式の入力文字列の少なくとも一部分を第2の形式に変換する。入力文字列の機械変換は、1つまたは複数のルールに従って翻訳可能な部分があるか否かを判断することと、翻訳されない任意の部分の書き直しを実行することを含む。入力文字列の機械変換については、以下で図2～図3に関してさらに詳しく説明する。

10

## 【0035】

システムは、変換された入力文字列に対して事後処理を実行する(112)。たとえば、事後処理を使用して、パイプラインの最初の部分が正しく処理されなかった語または属性をブラックリストに載せることができる。たとえば、変換された出力文字列の品質は、言語的理由により特定の地理的地域において異なりうる(たとえば、南アフリカには、英語、オランダ語およびドイツ語が組み合わさった名前が沢山あって、変換が難しくなっている)。その結果、変換された語、たとえば通り名をブラックリストに載せることが、そのデータが現在信頼できない場合、実行されうる。したがって、マップデータを提示する例では、それらの語は表示用に変換されることなく、代わりに元の入力文字列が使用される。同様に、シンガポールには中国語と英語の両方による通りが沢山あるので、それらの英語の通り名を中国語に翻訳する必要はない。

20

## 【0036】

システムは、第2の形式の変換された文字列を出力する(114)。たとえば、変換された文字列は、後で使用するために(たとえば、データベースまたは他の格納場所に)保存される。いくつかの実装形態では、変換された文字列は、対応するマップデータが表示のために要求されたときに取り出される地理的ラベルである。たとえば、中国の情報を識別する地理的データの英語の文字列を有するデータベースは、中国の提示マップにラベルを付ける際に使用するために、中国語の文字列に変換される。別の例では、変換された文字列は、生成された後にユーザに提示される。いずれの例でも、変換された文字列のうちの1つまたは複数の単独で(たとえば、変換された文書)、または他のデータとともに(たとえば、マップ情報)提示される。

30

## 【0037】

図2は、入力文字列の機械変換のための例示的な方法200の流れ図である。便宜上、方法200は、方法200を実行する、1つまたは複数のコンピューティングデバイスを含むシステムに関して説明する。

## 【0038】

システムは入力文字列を受信する(202)。入力文字列は、たとえば、第1の形式から第2の形式に変換される入力文字列の集合から受信される。いくつかの実装形態では、入力文字列は、たとえば図1で説明したように、受信される前に1つまたは複数の事前処理ステップを経ている。あるいは、入力文字列は事前処理なしに直接処理される。

40

## 【0039】

システムは、受信された入力文字列にルールのグループを適用する(204)。ルールは、特定の特徴タイプおよび特徴タイプが入力文字列と合致した場合に実行するアクションを識別する。システムは、たとえばルール階層に従って入力文字列に各ルールを適用する。特徴は、入力文字列における特徴のラベリングに基づいて、特定の特徴タイプと合致する。ルールは、所与の特徴タイプに関するルールと合致する1つまたは複数の語のパターンを有する。入力文字列が入力パターンと合致した場合、ルールは合致する。いくつかの実装形態では、所与の言語または書記体系に関して定義されたルールごとに、それぞれの言語または書記体系において1つまたは複数の出力パターンがある。入力文字列が入力パターンと合致した場合、システムは、ルールによって定義されたそれぞれの出力パターン

50



に従って、1つまたは複数の出力語を生成する。

【0040】

たとえば、下に示すのは、地理的特徴の英語(「en」)名を日本語(「ja」)、韓国語(「ko」)、キルギス語(「ky」)、ロシア語(「ru」)、簡体字(「zh-Hans」)、および繁体字(「zh-Hant」)に変換するための3つの例示的なルール構造である。これらの例示的なルールは、明確にするために単純化されている。

【0041】

【数5】

```
select < feature_type: "TYPE_PARK" >
```

10

```
  name < text: "(.+) Park" language: "en" >
```

```
  out < text: "${1} 公園" language: "ja" >
```

```
  out < text: "${1} 공원" language: "ko" >
```

```
  out < text: "${1} парки" language: "ky" >
```

```
  out < text: "Парк ${1}" language: "ru" >
```

```
  out < text: "${1} 公园" language: "zh-Hans" >
```

```
  out < text: "${1} 公園" language: "zh-Hant" >
```

20

```
select < feature_type: "TYPE_TOURIST_DESTINATION" >
```

```
  name < text: "(.+) Zoo" language: "en" >
```

```
  out < text: "${1} 動物園" language: "ja" >
```

```
  out < text: "${1} 동물원" language: "ko" >
```

```
  out < text: "${1} зоопарки" language: "ky" >
```

```
  out < text: "Зоопарк ${1}" language: "ru" >
```

```
  out < text: "${1} 动物园" language: "zh-Hans" >
```

30

```
  out < text: "${1} 動物園" language: "zh-Hant" >
```

```
select < feature_type: "TYPE_POLITICAL" >
```

```
  name < text: "(.+) " language: "en" >
```

```
  out < text: "${1}" language: "ja" >
```

```
  out < text: "${1}" language: "ru" >
```

```
  out < text: "${1}" language: "ko" >
```

```
  out < text: "${1}" language: "ky" >
```

```
  out < text: "${1}" language: "zh-Hans" >
```

```
  out < text: "${1}" language: "zh-Hant" >
```

40

【0042】

したがって、たとえば、特徴タイプ「park」に関連するルールは、英語入力パターン「(.+) Park」を有し、それにより、「park」の前に何らかのワイルドカードテキストを有する特徴タイプparkの入力テキスト文字列は、ルールと合致する。たとえば、カリフォルニア州サンフランシスコにある公園を指す入力文字列Lafayette Parkは、入力パターンと合致する。対応する出力パターンが複数の言語で提供される。

50

## 【0043】

特に、所与の入力文字列について、「Lafayette Park」に再び目を向けると、システムは合致を求めてルールของすべてを探索する(walk)。「Lafayette Park」の例では、入力文字列の特徴「Park」は特徴タイプ「TYPE\_PARK」を有するとラベルを付けられるので、第1のルールは合致する。ルールは、「TYPE\_PARK」のサブタイプ(不図示)(たとえば、「TYPE\_MUNICIPAL\_PARK」)も合致するように、特徴タイプの存在論(ontology)に従って構成される。加えて、入力文字列「Lafayette Park」は入力パターン「(.+) Park」と合致するので、第1のルールは合致する。

## 【0044】

システムは、ルールに基づいて入力文字列の1つまたは複数の語を翻訳すべきか否かを判断する(206)。特に、入力文字列が特定の特徴およびルールと合致している場合、システムは1つまたは複数の出力文字列を生成する。出力パターンは、1つまたは複数の言語における翻訳された語を含みうる。例示的なルールに示されているように、「out」条項は、出力言語について、出力を生成する方法を指定するパターンを含む。「Lafayette Park」の場合、ルールは複数の言語における出力パターンを識別し、各々は「Park」をそれぞれの言語に翻訳する。たとえば、日本語では、ルールの出力パターンは「out<text: “\${1}公園” language: “ja”>」であり、ここで\${1}は「Lafayette」、「公園」は日本語における語「park」の翻訳である。こうして、文字列の一部分が第2の形式に翻訳される一方、文字列の一部分は第1の形式にとどまる。

## 【0045】

対照的に、カリフォルニア州にある都市を指し、特徴タイプ「TYPE\_CITY」を有する入力文字列「Menlo Park」は、上記の例示的なルールの第1のルールと合致しない。使用するタイプ体系によれば、特徴タイプ「TYPE\_CITY」は「TYPE\_PARK」のサブタイプであるとは見なされないからである。第1のルールは合致していないので、システムは特徴タイプ「TYPE\_Tourist\_Destination」に関する第2のルールを試みるが、またしても成功しない。

## 【0046】

しかしながら、「TYPE\_CITY」は「TYPE\_POLITICAL」のサブタイプであるので、第3のルールが合致する。しかしながら、この場合、ルールと合致する文字列の出力パターンは翻訳を含まず、全体の出力\${1}は「Menlo Park」であるので、翻訳される文字列の部分はない。

## 【0047】

システムは、入力文字列の1つまたは複数の残りの語を書き直す(208)。特に、1つまたは複数の語を有する入力文字列について、合致ルールに従って翻訳されなかった語は書き直される。書き直しは、語を第1の形式から第2の形式にマッピングする。第2の形式は、同じ言語もしくは異なる言語の場合、または同じ書記体系もしくは異なる書記体系の場合がある。特定の書き直し技法については、以下で図3に関してより詳しく説明する。

## 【0048】

システムは、変換された文字列を出力する(210)。変換された文字列は、ルールの適用による書き直された語と翻訳された語との結び付きを含む。いくつかの実装形態では、複数の変換された文字列が、それぞれの言語または書記体系で出力される(たとえば、「Park」は日本語「公園」、韓国語

## 【0049】

## 【数6】

“공원”

## 【0050】

、ロシア語「Парк」に翻訳される)。いくつかの他の実装形態では、特定の出力言語または書記体系が、入力文字列に関して指定される(たとえば、英語から日本語)。

## 【0051】

上記の例では、入力文字列「Lafayette Park」はルールと合致しており、結果的にルールを適用して「Lafayette公園」となり、ここで「公園」は日本語における「Park」の翻訳である。次いで、語「Lafayette」がターゲット書記体系、この場合は日本語に書き直される。この例では、書き直しは「ラファイエット」を返す。これは「Park」の翻訳と結び付けられ、混合形式「ラファイエット公園」となる。

#### 【0052】

同様に、第2の例「Menlo Park」では、翻訳される語がなかったので、入力文字列全体が書き直される。日本語への書き直しの場合、結果として生じる出力は「メンロー・パーク」である。特に、特徴タイプの使用に起因して、都市「Menlo Park」は文字どおりに書き直され、一方で公園「Lafayette Park」は部分的に書き直され、部分的に翻訳されることに留意されたい。

10

#### 【0053】

別の例として、入力文字列「Geneva city hall」はルールに基づいて「city hall」の翻訳を識別する。次いで、システムは「Geneva」に対し書き換えを実行する。「Geneva」は、人間が入力した翻訳のデータベースに現れることがある(たとえば、図1のステップ106に関して上述したように、システムは、書き直される出力として「Geneva」の翻訳を使用する)。次いで変換された出力は、「city hall」の翻訳と結び付いた「Geneva」の翻訳となる。

#### 【0054】

同様のプロセスは他のタイプのデータに使用されうる。たとえば、特定の企業ルールに従って企業名が処理されうる。したがって、企業名「Foo Bar Inc.」の場合、ルールは「Inc」を、別の形式への「incorporated」の翻訳により識別する。この翻訳は、「Foo Bar」の書き直しと結び付けられ、結果的に混合型の変換された出力文字列が生じる。

20

#### 【0055】

図3は、入力文字列の語を書き直すための例示的な方法300の流れ図である。便宜上、方法300は、方法300を実行する、1つまたは複数のコンピューティングデバイスを含むシステムに関して説明する。

#### 【0056】

システムは、第2の形式に翻訳されない第1の形式の入力文字列の語をトークン化する(302)。たとえば、システムは、文字列を別個の単語に分割することによって、入力文字列の語をトークン化する。英語では、たとえば、これは分割ポイントとしてスペースを使用することによって行われる。他の言語、たとえば、中国語、日本語、および韓国語では、書記体系に応じて、単語間のスペースがないことから、これらの言語用の他のトークン化技法が必要である。上記の例「Menlo Park」では、トークンは[「Menlo」、「Park」]となる。

30

#### 【0057】

システムは各トークンを第2の形式(たとえば、第2の言語または書記体系)に書き直す(304)。特に、各トークンは別個に書き直される。上記の例では、トークン[Menlo、Park]は[メンロー、パーク]に別個に書き直される。異なる書き直し技法を使用して、個別のトークンまたは単語を書き直すことができる。例示的な書き直し技法については、以下で詳述する。

40

#### 【0058】

随意に、入力形式と出力形式(たとえば、特定の入力言語と出力言語)との組合せに応じて、システムは書き直されたトークンに追加のルールを適用する(306)。具体的には、これらの追加のルールは、入力形式および出力形式に固有の特定の言語的現象に対処する。たとえば、1つの言語的現象は外連声、またはフランス語の場合は特に「フランス語のリエゾン」と呼ばれる。この現象は、特定の単語が、1つまたは複数の他の単語を有する文脈に応じてどのように取り扱われるかに関係する。特に、これは、形態素または単語の境界で生じる様々な音韻過程に関係する。

#### 【0059】

50

たとえば、フランス語名「Maison des abricots」(アブリコットの家:house of apricots)について考える。問題は、フランス語の単語「des」(英語: of)が、後に続く単語に応じて異なって発音されることである。この発音の相違は、フランス語をいくつかの他の言語に書き直すときに顕著になる。後に続く単語が子音で始まる場合、フランス語の「des」は[de]と発音される、つまり、単語の最後の「s」は発音されない。一方、後に続く単語が母音で始まる場合、フランス語の「des」の発音は、有声の聞き取れる[z]音を含み、[dez]となる。さらに複雑なことに、フランス語を日本語に正しく書き直すには、[z]音は「des」の書き直しに付加されず、代わりに、後に続く単語の書き直しの前に付加される必要がある。したがって、[「Maison」、「des」、「abricots」]を書き直すとき、言語固有の書き直しルールの適用は、[「メゾン」、「デズ」、「アブリコー」]から[「メゾン」、「デ」、「ザブリコー」]、すなわち([

10

【0 0 6 0】

【数7】

"mezō"

【0 0 6 1】

、「dezu」、

【0 0 6 2】

【数8】

"aburikō"

20

【0 0 6 3】

]から [

【0 0 6 4】

【数9】

"mezō"

【0 0 6 5】

、「de」、

【0 0 6 6】

【数10】

"zaburikō"

30

【0 0 6 7】

])に変わる。

【0 0 6 8】

これらの追加のルールを適用するために、入力トークンと出力トークンの両方が使用される。加えて、連声ルールについて、連声を引き起こす単語の辞書を使用してルールを生成することができる。たとえば、フランス語を日本語に書き直す場合、「-s」で終わるフランス語の単語の全部ではなく一部のみのみ、それが日本語の書き直しに現れる範囲でリエゾンを引き起こす。

40

【0 0 6 9】

システムは書き直された出力文字列を出力する(308)。複数のトークンが書き直された場合、追加のルールの適用に基づいて語または語順を変更して書き直された出力を形成するために、最後の一連の語が結び付けられる。

【0 0 7 0】

一部分が翻訳された場合、上記の変換された出力文字列を生成するために、この書き直された部分が翻訳された部分に結合される。いくつかの実装形態では、結び付けは、(たとえば、英語の書き直しを結び付ける際に)書き直された語の間にスペースを追加することを含む。たとえば、翻訳されたトークン[「Menlo」、「Park」]は、「Menlo Park」を

50

提供するために各語の間にスペースを追加することによって結び付けられる。いくつかの他の実装形態では、結び付けるときに言語固有のハイフネーションルールが適用される。たとえば、ドイツ語では、特定のタイプの文字列におけるハイフネーション(たとえば、「Anna-Karenina-Park」)を必要とするつづり字法がある。さらにいくつかの他の実装形態では、結び付けは、書き直されたトークンの間に他の特徴を追加することを含む。たとえば、日本語の単語の間にカタカナの中点が追加されることがある。したがって、入力文字列「Menlo Park」の場合、日本語の書き直されたトークン[メンロー、パーク]は出力文字列「メンロー・パーク」になる。ルールは、単にスペースの代わりに中点を挿入するだけの場合よりも複雑になりうる。たとえば、日本語のカタカナの中点文字を挿入するのは、挿入ポイントの両側が実際に日本語のかな(ひらがなまたはカタカナ)表記のうちの1つである場合のみである。したがって、「Menlo Park Bridge」を[「メンロー」、「パーク」、「橋」]に変換した後、結び付けて「メンロー・パーク橋」が作られる。メンロー(「Menlo」)とパーク(「Park」)との間には、両方の文字列がカタカナ書記体系であるので、中点が挿入される。一方、パーク(「Park」)と橋(「bridge」)との間には、前者のみがカタカナで書かれ、後者が日本語の漢字書記体系で書かれているので、中点は挿入されない。

10

**【0071】**

様々な技法を使用して、個々のトークン(たとえば、個々の単語)を書き直すことができる。特に、書き直し中に、各トークンが例外辞書に照らしてチェックされる。例外辞書を使用して、発音、およびそれに応じて他の言語への書き直しが不規則である単語に対処する。例外辞書は様々なタイプのデータを含みうる。

20

**【0072】**

いくつかの実装形態では、例外辞書がログ分析(たとえば、検索ログ、マップログ)に基づいて実装され、これを使用して、どの文字列がユーザに頻繁に表示されるかを識別する。たとえば、マッピングサービスのユーザが閲覧した世界の特定の部分を識別し、それらのマップに提示される語をカウントすることが可能である。カウントは指定期間(たとえば、数ヵ月)に、全ユーザにわたって合算される。結果は、ユーザ要求元によって分けられる。したがって、異なる言語(たとえば、フランス語、英語、ドイツ語、イタリア語)における特定の文字列が、異なる地域、たとえば、日本、ロシア、中国、韓国のユーザに頻繁に示される。ソース言語の発音の不規則性に応じて、人間の評価者(たとえば、特定の言語のスキルを有する人、たとえば言語教師)が上位2,000~20,000単語の書き直しを手動で入力する。したがって、上位N個の最頻出単語は、正確性に関して高い信頼性を有する。これらの書き直しは例外辞書に追加される。

30

**【0073】**

いくつかの他の実装形態では、例外辞書が経験則を使用して実装され、特に不規則な文字列を識別する。たとえば、英語ではなくゲール語である可能性が高いグレートブリテン内の名前の文字列を識別する。次いで、これらのゲール語の単語は、英語から別の言語(たとえば、日本語)に異なるプロセスを使用して書き直される。次いで、これらの書き直しは例外辞書に追加される。

**【0074】**

トークンが例外辞書内で発見されなかった場合、特定の書き直し技法が選択される。たとえば、特定の書き直し技法が、ソースおよびターゲットの言語または書記体系に基づいて選択されうる。したがって、たとえば、ある書き直し技法がスペイン語入力を日本語出力に変換するとき使用され、別の書き直し技法が英語入力を日本語出力に変換するとき使用されうる。様々な書き直し技法が使用されうる。例示的な書き直し技法は、文脈依存書き換えルールに基づく単純な書き直し、またはテキスト音声技術に基づく複雑な書き直しを含むが、これらに限定されない。

40

**【0075】**

文脈依存書き換えルールに基づく単純な書き直しは、たとえば、一定程度の規則的な句読法を有する入力言語(たとえば、スペイン語またはチェコ語)を書き直すために使用され

50

る。特定のライブラリに従って語を書き直す際に使用するために、書き換えルールエンジンが受信されうる。書き直しライブラリを使用して、根底の単語を翻訳することなく、ある書記体系から別の書記体系に文字を変換する。たとえば、英語入力「bat」がギリシャ語の「βατ」に変換されうる。

【0076】

いくつかの実装形態では、オープンソース書き直しライブラリが使用される。ライブラリは、ある書記体系から別の書記体系に文字がどのように書き直されるかを定義するルールを含む。特定の変換ルールシンタックスに従って使用されるライブラリ用のカスタム変換ルールを生成することができる。

【0077】

いくつかの実装形態では、テキストは、入力言語から入力言語の音素表示(たとえば、国際音標文字(IPA)またはX-SAMPA表記)、入力言語の音素表示から出力言語の音素表示、出力言語の音素表示から出力言語の書記形式へと、複数のステップで変換される。

【0078】

ライブラリは、特定のタイプのテキストを書き直すためのルールの集合を含む。たとえば、場合によっては、ギリシャ語とラテン語の両方の文字間には、1対1の関係がある。ルールは、ソース文字列とターゲット文字列の間でマッピングする。以下はこの関係を示している: $\pi <> p$ 。ルールによれば、ギリシャ語からラテン語に書き直すときには、 $\pi$ をpに変換し、ラテン語からギリシャ語に書き直すときには、pを $\pi$ に変換する。より一般的には、シンタックスは、文字列1<>文字列2である。ルールはより複雑になることがあり、たとえば、ルールは文脈に基づくことがある。たとえば、ギリシャ語では、「γ」から「n」への変換を、それが以下の文字、すなわち、γ、κ、

【0079】

【数11】

ξ

【0080】

またはχのうちのいずれかの前に行うルールとなっている。それ以外の場合は「g」に変換するルールとなっている。同様に別の例では、ギリシャ語のシグマは、単語の最後にある(ただし、完全に別個ではない)場合には

【0081】

【数12】

“ς”

【0082】

と書かれ、それ以外の場合には「σ」と書かれる。文字をギリシャ語からラテン語に変換するときには、これは問題ではない。しかし、文字をラテン語からギリシャ語に戻すときには、これは問題であり、特にルールは、書き直される文字列の文脈に応じていつ「s」に変換すべきかを定義している。

【0083】

たとえば、チェコ語の入力文字列を韓国語の入力文字列に変換するために書き直しを実行するとき、以下の連鎖を使用できる。チェコ語→IPA記号を使用した音素のチェコ語→ラテン語表記の韓国語→韓国語表記の韓国語。特に、これは以下の変換を行う。

【0084】

【数13】

hořký

【0085】

→

【0086】

10

20

30

40

50

【数 1 4】

h orki :

【0 0 8 7】

→holeusyuki→

【0 0 8 8】

【数 1 5】

호르슈키

【0 0 8 9】

10

これらのステップは分離可能であり、それにより、ルールの中のいくつかの部分を独立して使用し、また他のルールにつないで、複数の言語ペア全体で再利用することができる。たとえば、チェコ語を日本語に書き直すためのルールは、チェコ語から韓国語へのルールとまったく同じ「IPA記号を使用した音素のチェコ語」において機能する。したがって、チェコ語から「IPA記号を使用した音素のチェコ語」への変換は、韓国語を生成するときと、日本語出力を生成するときの両方に使用できる。一例として、以下はチェコ語の文字nを文脈に応じて[n]または[n̥]のいずれかに変換するためのルールである。

{n} [g k q x] →

【0 0 9 0】

【数 1 6】

20

ŋ

【0 0 9 1】

; #[g k q x]のうちの1つの前では、チェコ語の書記素nは、軟口蓋鼻音

【0 0 9 2】

【数 1 7】

ŋ

【0 0 9 3】

と発音される。

30

{n}→n; #他の文脈では、チェコ語の書記素nは、n音として発音される。

【0 0 9 4】

また、一例として、これらは、文脈に応じて、チェコ語の文字

【0 0 9 5】

【数 1 8】

ř

【0 0 9 6】

(IPA表記では

【0 0 9 7】

【数 1 9】

40

[r]

【0 0 9 8】

と発音される)を異なる韓国語-ラテン語文字列に変換するためのルールである。

【0 0 9 9】

【数 2 0】

{r} \$end → leusi; #ř

【0 1 0 0】

50

が単語の末尾、たとえば、

【0 1 0 1】

【数 2 1】

$kou\check{r} \rightarrow \text{코우르시}$

【0 1 0 2】

【数 2 2】

$\{r_i\} \$vowel \rightarrow leuj; \# \check{r}$

【0 1 0 3】

10

が母音の前

【0 1 0 4】

【数 2 3】

$\{r_i\} k \rightarrow leusyu; \# \check{r}$

【0 1 0 5】

がk音の前

【0 1 0 6】

【数 2 4】

$\{r_i\} \rightarrow leuju; \# \check{r}$

20

【0 1 0 7】

が任意の他の文脈。

【0 1 0 8】

個々の語の書き直しを実行するとき、入力形式および出力形式に基づいて特定のルールを適用するために適切なライブラリを使用する。次いで、書き直された語を上記のようにさらに処理して、第1形式の入力文字列から第2の形式の出力文字列への変換である最終出力文字列を生成することができる。

【0 1 0 9】

不規則な発音が多い入力言語、特に英語の場合、テキスト音声技術を使用する一層複雑なシステムが使用されうる。具体的には、入力単語を出力単語または中間形式に変換するために、例に基づいてオフラインで訓練される幅広い種類のモデルが使用されうる。いくつかの実装形態では、システムはモデルの種類として有限状態トランスデューサを使用する。一例では、モデルが入力単語および対応する書き直しの例に対して直接訓練されうる。実行時に、入力単語が、訓練されたモデルによって多数の可能な出力形式に変換される。システムは、有限状態トランスデューサまたは他のモデルによって割り振られた数字の重み付けに基づいて、単一の最も優れた(または代替的にn番目に優れた)出力形式を導出する。

30

【0 1 1 0】

別の例では、変換全体は複数のステップを含みうる。入力形式は最初に中間形式に変換される。中間変換がまったく実行されないこと、または任意の回数だけ実行されることがある。出力書き直しが中間形式のうちの1つまたは複数から、場合により入力単語と組み合わせ導出される。いくつかの実装形態では、発音表記が中間形式として使用される。特に、入力単語が最初にソース言語の発音表記に変換される。次いで、この発音表記は、ターゲット言語の発音表記に変換される。次いで、ターゲット言語の発音表記は、ターゲット言語の正字法による形式にさらに変換される。あるいは、入力単語はターゲット言語の発音表記または正字法による表記に変換されうる。いくつかの実装形態では、システムは有限状態トランスデューサに支援されて計算されるソース言語とターゲット言語の両方における発音表記を使用して書き直しを実行する。

40

【0 1 1 1】

50



図4は、例示的なシステムアーキテクチャ400である。システムアーキテクチャ400は、入力文字列を第1の形式から第2の形式に変換するための動作を実行することが可能である。システムアーキテクチャ400は、1つまたは複数のプロセッサ402(たとえば、IBM PowerPC、Intel Pentium(登録商標) 4など)、1つまたは複数の随意のディスプレイデバイス404(たとえば、CRT、LCD)、グラフィックス処理ユニット406(たとえば、NVIDIA GeForceなど)、ネットワークインターフェース408(たとえば、イーサネット(登録商標)、FireWire、USBなど)、随意の入力デバイス410(たとえば、キーボード、マウスなど)、および1つまたは複数のコンピュータ可読媒体412を含む。これらの構成要素は、1つまたは複数のバス414(たとえば、EISA、PCI、PCI Expressなど)を使用して通信およびデータを交換する。

#### 【0112】

「コンピュータ可読媒体」という用語は、実行のためにプロセッサ402に命令を提供することに参加する任意の媒体を指す。コンピュータ可読媒体412は、オペレーティングシステム416(たとえば、Mac OS(登録商標)、Windows(登録商標)、Linux(登録商標)など)、ネットワーク通信モジュール418、文字列変換器420、および他のアプリケーション424をさらに含む。

#### 【0113】

オペレーティングシステム416は、マルチユーザ、マルチ処理、マルチタスク、マルチスレッド、リアルタイムなどでありうる。オペレーティングシステム416は、限定はしないが、入力デバイス410からの入力を認識すること、ディスプレイデバイス404に出力を送ること、コンピュータ可読媒体412のファイルおよびディレクトリ(たとえば、メモリまたは記憶デバイス)を追跡すること、周辺デバイス(たとえば、ディスクドライブ、プリンタなど)を制御すること、および1つまたは複数のバス414のトラフィックを管理することを含む基本的タスクを実行する。ネットワーク通信モジュール418は、ネットワーク接続を確立し維持するための様々な構成要素(たとえば、TCP/IP、HTTP、イーサネット(登録商標)などの通信プロトコルを実施するためのソフトウェア)を含む。

#### 【0114】

文字列変換器420は、図1～図3に関して説明したように入力文字列を第1の形式から第2の形式に変換するための様々な機能を実行するための様々なソフトウェア構成要素を提供する。

#### 【0115】

本明細書で説明する対象および動作の実施形態は、デジタル電子回路において、または本明細書で開示される構造およびその構造の同等物を含む、コンピュータソフトウェア、ファームウェアもしくはハードウェアにおいて、またはこれらのうちの1つもしくは複数の組合せで実施できる。本明細書で説明する対象の実施形態は、1つまたは複数のコンピュータプログラム、すなわち、データ処理装置による実行のため、またはデータ処理装置の動作を制御するため、コンピュータ記憶媒体に符号化されたコンピュータプログラム命令の1つまたは複数のモジュールとして実施できる。代替的にまたは追加として、プログラム命令は、人工生成の伝搬信号、たとえば、データ処理装置による実行のために適切な受信機装置に送信する情報を符号化するために生成される、機械生成の電気信号、光信号または電磁信号に符号化できる。コンピュータ記憶媒体は、コンピュータ可読記憶デバイス、コンピュータ可読記憶基板、ランダムもしくは順次アクセスメモリアレイもしくはデバイス、またはこれらのうちの1つもしくは複数の組合せであること、またはこれらに含まれることがある。さらに、コンピュータ記憶媒体は伝搬信号ではないが、コンピュータ記憶媒体は、人工生成の伝搬信号に符号化されたコンピュータプログラム命令の発信元または宛先でありうる。コンピュータ記憶媒体はまた、1つまたは複数の別個の物理構成要素または媒体(たとえば、複数のCD、ディスク、または他の記憶デバイス)であること、またはこれらに含まれることがある。

#### 【0116】

本明細書で説明する動作は、1つまたは複数のコンピュータ可読記憶デバイスに保存されているか、他のソースから受信したデータに対してデータ処理装置によって実行される

10

20

30

40

50

動作として実施できる。

#### 【0117】

「データ処理装置」という用語は、たとえばプログラマブルプロセッサ、コンピュータ、チップ上のシステム、またはこれらの複数のもしくは組合せを含む、データを処理するためのあらゆる種類の装置、デバイス、および機械を包含する。装置は、専用論理回路、たとえばFPGA(フィールドプログラマブルゲートアレイ)またはASIC(特定用途向け集積回路)を含むことができる。この装置はまた、ハードウェアに加えて、問題のコンピュータプログラムの実行環境を作るコード、たとえば、プロセッサファームウェア、プロトコルスタック、データベース管理システム、オペレーティングシステム、クロスプラットフォーム実行時環境、仮想機械またはこれらのうちの1つもしくは複数の組合せを構成するコードを含むことができる。装置および実行環境は、ウェブサービス、分散コンピューティング、グリッドコンピューティングインフラストラクチャなど様々な異なるコンピューティングモデルインフラストラクチャを実現することができる。

10

#### 【0118】

(プログラム、ソフトウェア、ソフトウェアアプリケーション、スクリプトまたはコードとしても知られている)コンピュータプログラムは、コンパイルまたは翻訳された言語、宣言形または手続き形の言語を含む任意の形式のプログラミング言語で書かれてよく、独立プログラムとしての形式、またはモジュール、構成要素、サブルーチン、オブジェクトもしくはコンピューティング環境での使用に適した他のユニットとしての形式など、任意の形式で展開されてよい。コンピュータプログラムは、ファイルシステム内のファイルに対応しうるが、対応する必要はない。プログラムは、他のプログラムもしくはデータ(たとえば、マーク付け言語文書に記憶される1つもしくは複数のスクリプト)を保持するファイルの一部分、問題のプログラム専用の1つのファイル、または複数の調整されたファイル(たとえば、1つもしくは複数のモジュール、サブプログラムもしくはコードの一部分を記憶するファイル)に記憶できる。コンピュータプログラムは、1つのコンピュータ、または1つの場所に位置するか、もしくは複数の場所に分布し、通信ネットワークで相互接続された複数のコンピュータで実行されるように展開できる。

20

#### 【0119】

本明細書で説明するプロセスおよび論理フローは、入力データに対し動作すること、および出力を生成することによってアクションを実行するための1つまたは複数のコンピュータプログラムを実行する1つまたは複数のプログラマブルプロセッサによって実行できる。プロセスおよび論理フローはまた、専用論理回路、たとえばFPGA(フィールドプログラマブルゲートアレイ)またはASIC(特定用途向け集積回路)によって実行可能であり、装置はまた、専用論理回路、たとえばFPGAまたはASICとして実装可能である。

30

#### 【0120】

コンピュータプログラムの実行に適したプロセッサは、たとえば、汎用および専用のマイクロプロセッサ、ならびに任意の種類のデジタルコンピュータのうちの任意の1つまたは複数のプロセッサを含む。一般に、プロセッサは、読取り専用メモリまたはランダムアクセスメモリまたはその両方から命令およびデータを受信する。コンピュータの基本的要素は、命令に従いアクションを実行するためのプロセッサ、ならびに命令およびデータを記憶するための1つまたは複数のメモリデバイスである。一般に、コンピュータはまた、データを記憶するための1つもしくは複数の大容量記憶デバイス、たとえば磁気ディスク、光磁気ディスクもしくは光ディスクを含み、またはかかる大容量記憶デバイスからデータを受信し、もしくはかかる大容量記憶デバイスにデータを転送し、もしくはその両方を行うように動作可能に結合される。ただし、コンピュータはかかるデバイスを備えている必要はない。さらに、コンピュータは、別のデバイス、いくつか挙げてみると、たとえば携帯電話、携帯情報端末(PDA)、携帯音声もしくは映像プレーヤー、ゲームコンソール、全地球測位システム(GPS)受信機、またはポータブル記憶デバイス(たとえば、ユニバーサルシリアルバス(USB)フラッシュドライブ)などに組み込まれうる。コンピュータプログラム命令およびデータを記憶するのに適したデバイスは、あらゆる形式の不揮発性メモリ、

40

50

媒体およびメモリデバイス、たとえば、半導体メモリデバイス(たとえば、EPROM、EEPROM およびフラッシュメモリデバイス)、磁気ディスク(たとえば、内部ハードディスクまたは取り外し可能ディスク)、光磁気ディスクならびにCD-ROMおよびDVD-ROMディスクを含む。プロセッサおよびメモリは専用論理回路によって補完され、または専用論理回路に組み込まれてよい。

#### 【0121】

ユーザと対話するために、本明細書で説明する対象の実施形態は、ユーザに情報を表示するためのディスプレイデバイス、たとえば、CRT(ブラウン管)またはLCD(液晶ディスプレイ)モニタ、ならびにユーザがコンピュータに入力を行うことができるキーボードおよびポインティングデバイス、たとえば、マウスまたはトラックボールを有するコンピュータで実施できる。他の種類のデバイスを使用してユーザと対話することも可能で、たとえば、ユーザに提供されるフィードバックは、任意の形式の知覚フィードバック、たとえば、視覚フィードバック、聴覚フィードバックまたは触覚フィードバックであってよく、ユーザからのインプットは、音響、音声または触覚インプットを含む任意の形式で受信できる。また、コンピュータは、ユーザによって使用されているデバイスにドキュメントを送り、かかるデバイスからドキュメントを受信することによって、たとえばユーザのクライアントデバイス上のウェブブラウザに、ウェブブラウザから受信したリクエストに応答してウェブページを送ることによって、ユーザと対話することができる。

10

#### 【0122】

本明細書で説明する対象の実施形態は、たとえばデータサーバとしての後置構成要素を含む、またはミドルウェア構成要素、たとえばアプリケーションサーバを含む、または前置構成要素、たとえば、本明細書で説明する対象の実装形態とユーザが対話できるグラフィカルユーザインターフェースもしくはウェブブラウザを有するクライアントコンピュータを含む、または1つもしくは複数のかかる後置、ミドルウェアもしくは前置構成要素の任意の組合せを含むコンピューティングシステムで実施できる。システムの構成要素は、デジタルデータ通信の任意の形式または媒体、たとえば通信ネットワークによって相互接続できる。通信ネットワークの例には、ローカルエリアネットワーク(「LAN」)、広域ネットワーク(「WAN」)、インターネットワーク(たとえば、インターネット)、およびピアツーピアネットワーク(たとえば、アドホックピアツーピアネットワーク)が含まれる。

20

#### 【0123】

コンピューティングシステムは、クライアントおよびサーバを含むことができる。クライアントおよびサーバは、一般に互いに離れており、通常は通信ネットワークを介して対話する。クライアントとサーバとの関係は、それぞれのコンピュータで稼動し、互いにクライアント-サーバ関係を有するコンピュータプログラムによって生じる。いくつかの実施形態では、サーバは(たとえば、クライアントデバイスと対話しているユーザにデータを表示し、かかるユーザからユーザ入力を受信するために)クライアントデバイスにデータ(たとえば、HTMLページ)を送信する。(たとえば、ユーザ対話の結果)クライアントデバイスで生成されたデータは、サーバにおいてクライアントデバイスから受信されうる。

30

#### 【0124】

本明細書は多くの特定の实施詳細を含んでいるが、かかる詳細を、いずれかの発明または特許請求内容の範囲の限定と解釈すべきではなく、特定の発明の特定の实施形態に固有の特徴の説明と解釈すべきである。また、個別の实施形態に関して本明細書において説明するいくつかの特徴は、1つの实施形態において組合せで実施できる。反対に、1つの实施形態に関して説明する様々な特徴は、複数の实施形態で個別に、または任意の適切な副組合せで実施できる。さらに、特徴は一定の組合せで機能するものとして上述され、当初はそういうものとして特許請求されることもあるが、特許請求される組合せによる1つまたは複数の特徴は、場合によっては、当該組合せにより実施可能であり、特許請求される組合せは副組合せまたは副組合せの変形を対象にしよう。

40

#### 【0125】

同様に、動作は特定の順序で図面に示されているが、これについては、所望の結果を達

50

成するために、そのような動作を示された特定の順序でもしくは順次に実行すること、またはすべての示された動作を実行することを要求するものとして理解すべきではない。状況によっては、マルチタスク処理および並列処理が有利なこともある。また、上述の実施形態における様々なシステム構成要素の分離については、すべての実施形態でそのような分離を要求するものとして理解すべきではなく、説明されるプログラム構成要素およびシステムは一般に1つのソフトウェア製品への統合、または複数のソフトウェア製品へのパッケージ化が可能であると理解すべきである。

【0 1 2 6】

以上、対象の特定の実施形態について説明してきた。他の実施形態も、後述の特許請求の範囲内にある。場合によっては、特許請求の範囲で列挙されるアクションは、異なる順序で実行可能であり、その場合でも所望の結果を達成できる。また、添付の図に記載のプロセスは、所望の結果を達成するために、必ずしも示された特定の順序または順次であることを要求しているわけではない。いくつかの実装形態では、マルチタスク処理および並列処理が有利なこともある。

10

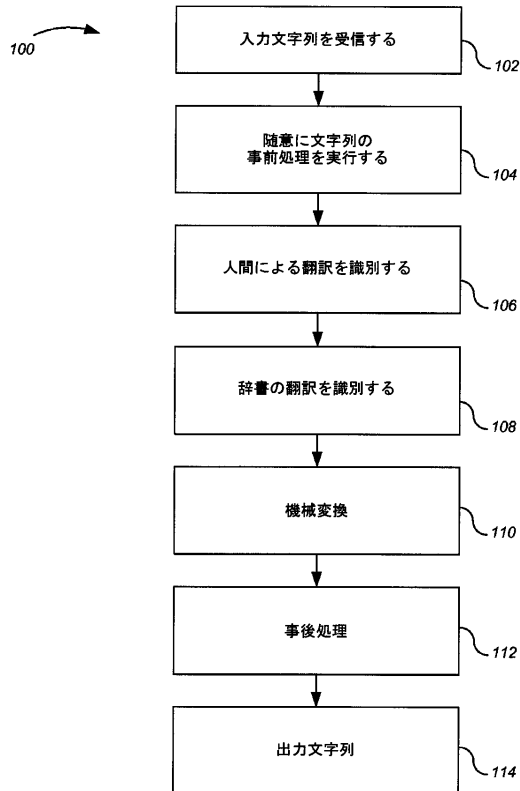
【符号の説明】

【0 1 2 7】

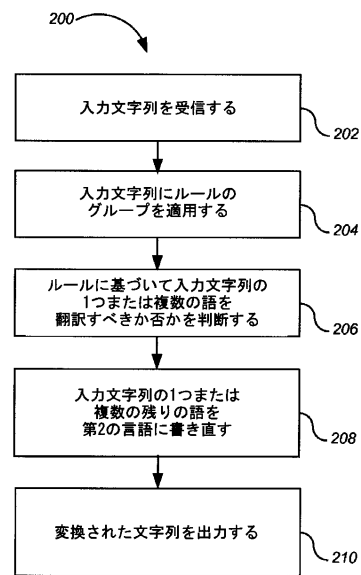
- 400 システムアーキテクチャ
- 402 プロセッサ
- 404 ディスプレイデバイス
- 406 グラフィックス処理ユニット
- 408 ネットワークインターフェース
- 410 入力デバイス
- 412 コンピュータ可読媒体
- 414 バス
- 416 オペレーティングシステム
- 418 ネットワーク通信モジュール
- 420 文字列変換器
- 424 他のアプリケーション

20

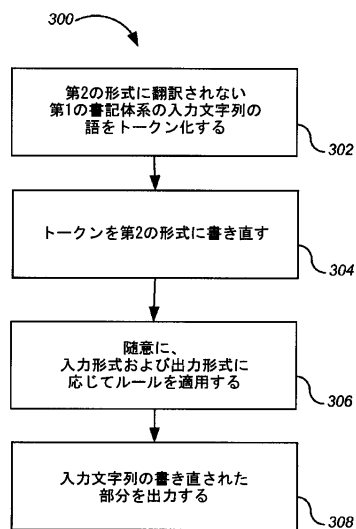
【図 1】



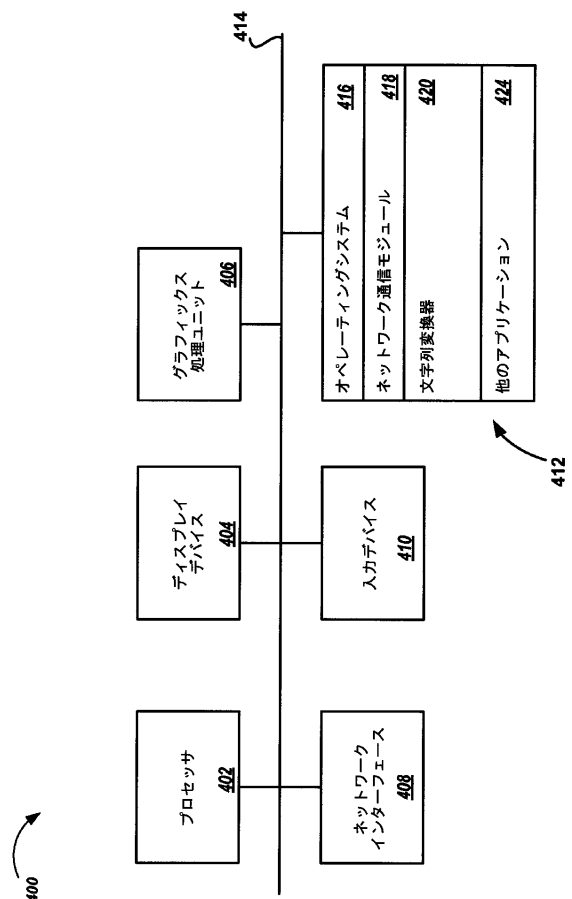
【図 2】



【図 3】



【図 4】



---

フロントページの続き

- (72)発明者 マルティン・ヤンシェ  
アメリカ合衆国・ニューヨーク・10032・ニュー・ヨーク・フォート・ワシントン・アヴェニ  
ュー・280・アパート・62
- (72)発明者 リチャード・スプロート  
アメリカ合衆国・オレゴン・97201・ポートランド・サウスウエスト・ジェファーソン・スト  
リート・1030・アパート・252
- (72)発明者 竹中 浩  
千葉県浦安市富士見4-6-38-603
- (72)発明者 寺島 有為  
東京都目黒区緑ヶ丘2-19-29 #102

合議体

審判長 佐藤 智康

審判官 相崎 裕恒

審判官 宮久保 博幸

- (56)参考文献 特開平10-312382 (JP, A)  
特開平8-328585 (JP, A)  
米国特許出願公開第2009/0144049 (US, A1)  
特開2005-216127 (JP, A)  
特表2005-529386 (JP, A)

- (58)調査した分野(Int.Cl., DB名)  
G06F17/20-28