

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5535417号
(P5535417)

(45) 発行日 平成26年7月2日(2014.7.2)

(24) 登録日 平成26年5月9日(2014.5.9)

(51) Int.Cl.

F I

G O 6 F 17/21 (2006.01)
G O 6 F 17/27 (2006.01)

G O 6 F 17/21 5 9 2 F
G O 6 F 17/21 5 9 2 J
G O 6 F 17/27 X

請求項の数 3 (全 21 頁)

(21) 出願番号	特願2001-536716 (P2001-536716)	(73) 特許権者	500046438
(86) (22) 出願日	平成12年10月13日 (2000.10.13)		マイクロソフト コーポレーション
(65) 公表番号	特表2003-514304 (P2003-514304A)		アメリカ合衆国 ワシントン州 9805
(43) 公表日	平成15年4月15日 (2003.4.15)		2-6399 レッドモンド ワン マイ
(86) 国際出願番号	PCT/US2000/028486		クロソフト ウェイ
(87) 国際公開番号	W02001/035250	(74) 代理人	100107766
(87) 国際公開日	平成13年5月17日 (2001.5.17)		弁理士 伊東 忠重
審査請求日	平成19年10月15日 (2007.10.15)	(74) 代理人	100070150
審判番号	不服2011-16022 (P2011-16022/J1)		弁理士 伊東 忠彦
審判請求日	平成23年7月25日 (2011.7.25)	(74) 代理人	100091214
(31) 優先権主張番号	60/163,902		弁理士 大貫 進介
(32) 優先日	平成11年11月5日 (1999.11.5)	(72) 発明者	カイーフ リー
(33) 優先権主張国	米国 (US)		アメリカ合衆国 98072 ワシントン
(31) 優先権主張番号	09/606,660		州 ウッディンビル 214 ウェイ ノ
(32) 優先日	平成12年6月28日 (2000.6.28)		ースイースト 14233
(33) 優先権主張国	米国 (US)		最終頁に続く

(54) 【発明の名称】 スペルミス、タイプミス、および変換誤りに耐性のある、あるテキスト形式から別のテキスト形式に変換する言語入力アーキテクチャ

(57) 【特許請求の範囲】

【請求項 1】

言語処理方法であって、
ユーザインタフェースを介して、入力文字列を受信するステップであって、前記入力文字列は表音テキストを含み、前記表音テキストはピンインであり、前記入力文字列は、タイピング誤りを含む、ステップと、
前記入力文字列に対する複数のタイピング候補のリストを作成するステップであって、前記複数のタイピング候補のリストは、タイピング候補が前記入力文字列として誤って入力された確率を示す第1の確率に基づいて、タイピング候補となりうる候補データを有する候補データベースから前記複数のタイピング候補を選択することにより作成され、前記リスト内の各タイピング候補は、異なる複数のセグメントに分割され、前記第1の確率は、複数のユーザから収集されたトレーニングデータを使用してトレーニングされる、ステップと、
前記リスト内の前記複数のタイピング候補の中から、前記入力文字列と置き換えるためのタイピング候補を決定し、決定されたタイピング候補に対応する言語テキストの変換候補を決定するステップであって、前記タイピング候補は、所与のセンテンス内に前記タイピング候補の各セグメントが存在しうる確率を示す第2の確率に基づいて、前記リスト内の個々のタイピング候補の複数のセグメントの中から最高確率となる1つまたは複数のセグメントを各々選択することによって決定され、当該決定されたタイピング候補は前記選択されたセグメントを含み、前記言語テキストは、漢字である、ステップと、

10

20

前記決定された変換候補を用いて、前記入力文字列を出力文字列に変換するステップとを備えることを特徴とする言語処理方法。

【請求項 2】

コンピュータに請求項 1 記載の方法を実行させることが可能な命令を有するコンピュータプログラムを記録したコンピュータ読取り可能な記録媒体。

【請求項 3】

言語処理システムであって、

プロセッサと、

前記プロセッサにより処理される、コンピュータ実行可能な命令を記憶するメモリとを備え、

前記コンピュータ実行可能な命令は、前記プロセッサに、

ユーザインタフェースを介して入力文字列を受信するステップであって、前記入力文字列は表音テキストを含み、前記表音テキストはピンインであり、前記入力文字列は、タイピング誤りを含む、ステップと、

前記入力文字列に対する複数のタイピング候補のリストを作成するステップであって、前記複数のタイピング候補のリストは、タイピング候補が前記入力文字列として誤って入力された確率を示す第 1 の確率に基づいて、タイピング候補となりうる候補データを有する候補データベースから前記複数のタイピング候補を選択することにより作成され、前記リスト内の各タイピング候補は、異なる複数のセグメントに分割され、前記第 1 の確率は、複数のユーザから収集されたトレーニングデータを使用してトレーニングされる、ステップと、

前記リスト内の前記複数のタイピング候補の中から、前記入力文字列と置き換えるためのタイピング候補を決定し、決定されたタイピング候補に対応する言語テキストの変換候補を決定するステップであって、前記タイピング候補は、所与のセンテンス内に前記タイピング候補の各セグメントが存在しうる確率を示す第 2 の確率に基づいて、前記リスト内の個々のタイピング候補の複数のセグメントの中から最高確率となる 1 つまたは複数のセグメントを各々選択することによって決定され、当該決定されたタイピング候補は前記選択されたセグメントを含み、前記言語テキストは、漢字である、ステップと、

前記決定された変換候補を用いて、前記入力文字列を出力文字列に変換するステップとを実行させることを特徴とする言語処理システム。

【発明の詳細な説明】

【0001】

(技術分野)

本発明は、言語入力方法およびシステムに関する。より詳細には、本発明は、テキスト入力時に発生するタイプミスおよびある言語形式から別の言語形式への変換時に発生する変換誤りの両方に対する誤り耐性 (error tolerance) のある言語入力方法およびシステムを実現する。

【0002】

(発明の背景)

言語固有のワードプロセッサソフトが何年もの間存在している。より複雑なワードプロセッサは、スペルおよび文法の訂正など、ユーザに高等なツールを提供し、文書を起草する際に補助する。例えば、多数のワードプロセッサは、スペルミスの単語や文法的に間違っているセンテンス構造を識別し、場合によっては、識別された誤りを自動的に訂正することができる。

【0003】

一般に、誤りがテキストに入り込む原因は 2 つある。原因の 1 つは、ユーザが単に、正しいスペルまたはセンテンス構造を知らないことである。ワードプロセッサは、ユーザが正しいスペルまたは言葉使いを選択する際に手助けするために提案することができる。第 2 のより一般的な誤りの原因は、ユーザが、正しいスペルまたは文法構造を知っていても単語またはセンテンスを間違ってコンピュータに入力することである。このような状況で

10

20

30

40

50

は、ワードプロセッサは、不適切に入力された文字列を識別し、意図した単語または語句に訂正するのに大変役立つことが多い。

【 0 0 0 4 】

入力誤りは、欧文文字を使用しない言語用に設計されたワードプロセッサで起こりがちである。英語バージョンの Q W E R T Y キーボードなど言語固有のキーボードは、多くの言語の場合、そのような言語がキーボードのキーとして使いやすく配列できるよりも多くの文字を有するため、存在していない。例えば、多くのアジア言語は数千文字を含む。これだけ多くの様々な文字について別々のキーをサポートするキーボードを構築することは実際上不可能である。

【 0 0 0 5 】

費用のかかる言語および方言固有のキーボードを設計するのではなく、言語固有の文書処理システムは、ユーザが小さな文字セットキーボード（例えば、Q W E R T Y キーボード）から表音テキスト（*phonetic text*）を入力し、その表音テキストを言語テキストに変換することができる。「表音テキスト」は、所与の言語を発話したときに発生する音を表すが、「言語テキスト」はテキストで表示されたとき実際に書かれた文字を表す。例えば、中国語では、ピンインは表音テキストの一例であり、漢字は言語テキストの一例である。表音テキストを言語テキストに変換することにより、多くの様々な言語を、従来のコンピュータおよび標準 Q W E R T Y キーボードを使用して言語固有のワードプロセッサによって処理できる。

【 0 0 0 6 】

表音入力を必要とするワードプロセッサでは、2種類の入力誤りが発生する可能性がある。1つの種類の誤りは、一般的なタイプミスである。しかし、テキストにタイプミスがないとしても、文書処理エンジンが表音テキストを意図しない文字テキストに誤って変換するという誤りもありえる。これら2つの問題の両方が、同じ表音テキスト入力文字列で作用した場合、一連の複数の誤りが生じることがある。状況によっては、語句またはセンテンスの文脈全体を時間をかけて調査しなければ、タイピングで生じる誤りが容易に追跡できない場合がある。

【 0 0 0 7 】

本明細書で説明する本発明は、表音テキストをタイプした場合のユーザによる前者のタイプの入力誤りを主に対象としているが、文書処理エンジンで生じた変換誤りに対する耐性も提供する。このようなタイピング誤りと関連する問題をよく説明するために、表音テキストであるピンインを言語テキストである漢字に変換する中国語ベースのワードプロセッサを考察する。

【 0 0 0 8 】

表音テキストを入力すると多くの場合タイピング誤りが増える理由はいくつかある。理由の1つは、英語用のキーボードでの平均タイピング精度は英語を話す国々の場合よりも中国の場合の方が低いということである。第2の理由は、表音はそれほど頻繁に使用されているわけではないということである。初期の教育期間中、ユーザは、例えば、英語を話すユーザが英語で単語を綴ることを教わるほど、表音綴りを勉強、学習する傾向はない。

【 0 0 0 9 】

表音テキスト入力中のタイピング誤りが増える第3の理由は、多くの人々が標準語ではなく方言を母語として話すというものである。表音テキストの起源である標準語は、第2言語である。特定の方言およびアクセントでは、話し言葉は、対応する適切な表音と一致しない場合があり、従って、ユーザが表音テキストをタイプすることをより難しくする。例えば、多くの中国人はさまざまな中国語方言を自分の第1言語として話しており、ピンインの起源である北京語を第2言語として教えられている。例えば、いつかの中国語方言では、一定の文脈で「h」と「w」を発音する際に区別がなく、他の方言では、「ng」と「n」についても同じことが言え、さらに他の方言では、「r」は明瞭に発音されない。そのため、北京語を第2言語として話す中国人ユーザは、ピンインを入力しようとしてタイピング誤りをしがちである。

10

20

30

40

50

【 0 0 1 0 】

タイピング誤りが増える理由としては他に、表音テキストをタイプしながら誤りをチェックすることが困難であるということが考えられる。これは、1つには、表音テキストが長く判読しにくくなり、読むことが困難になる傾向があるということによる。見たものがタイプしたものである英語ベースのテキスト入力と対照的に、表音テキストの入力は、「見たものが得られるもの」でないことが多い。むしろ、ワードプロセッサは、表音テキストを言語テキストに変換する。そのため、ユーザは一般に、表音テキストに誤りがないか調べることをしないが、表音テキストが言語テキストに変換されるまで多少待つ。

【 0 0 1 1 】

この最後の理由により、タイピング誤りは、ピンイン入力の文脈ではきわめて厄介なものである。ピンイン文字列は、文字間にスペーシングがないため再検討、訂正が非常に難しい。その代わりに、ピンイン文字は、ピンイン文字で構成される単語の数に関係なく混在する。さらに、ピンイン - 漢字変換はすぐに実行されることはないが、追加ピンインを入力するにつれ正しい解釈が定式化され続ける。従って、ユーザが間違ったピンイン記号をタイプした場合、単一の誤りが変換プロセスによって複合され、下流に伝播して、いくつかの追加誤りの原因となる場合がある。その結果、システムが漢字に確定変換しその後ユーザが誤りのあったことに気づくまでに、1回訂正するのにユーザは数回バックスペースを入力せざるをえないため、誤り訂正の時間が長くなる。システムによっては、元の誤りを明らかにできない場合さえある。

【 0 0 1 2 】

表音入力中に間違いが頻繁に生じることが予想されるため、表音入力の誤りを許容できるシステムが必要である。システムは、表音列がわずかに誤りのある文字列を含むとしても正しい答えを返すのが望ましい。

【 0 0 1 3 】

言語固有ワードプロセッサは他の問題に直面しており、これは、入力問題とは別であって、異なる言語から同じテキストに単語を入力するために2つの言語間でモードを切り替えることに関するものである。例えば、技術用語（例えば、Internet）などの英単語と翻訳が難しい用語（例えば、頭字語、記号、サーネーム、社名など）を含む文書草稿を中国語で作成することがよくある。従来のワードプロセッサでは、ユーザは異なる単語を入力するときに、一方の言語を他方の言語にモード切替をする必要がある。そのため、ユーザが異なる言語から単語を入力したい場合、ユーザはテキスト入力についての思考を停止し、モードをある言語から別の言語に切り替えて、単語を入力し、それからモードを最初の言語に戻す必要がある。このため、ユーザのタイピング速度が著しく低下し、ユーザは自分の注意をテキスト入力作業と、言語モードの変更という異質な制御作業との間で切り替える必要がある。

【 0 0 1 4 】

従って、モード切替を必要としない「モードレス」システムが必要なのである。モードを無用とするために、システムは、タイプしている言語を検出し、文字シーケンスを1単語ずつ一方の言語または他方の言語に動的に変換することができなければならない。

【 0 0 1 5 】

しかし、これは、両方の文脈で多くの文字列が適切な場合があるため、思ったほど簡単ではない。例えば、多くの有効な英単語はさらに有効なピンイン文字列でもある。さらに、ピンイン入力時に中国語文字間、および中国語文字と英単語との間にスペースがないため、曖昧さが増すことがある。

【 0 0 1 6 】

例えば、ユーザがピンイン入力テキスト「w o s h i y i g e z h o n g g u o r e n」をタイプすると、システムはこの文字列を中国語文字（一般に「私は中国人である」と翻訳される）に変換する。

【 0 0 1 7 】

ときには、「w o s h i y i g e z h o n g g u o r e n」とタイプする代わりに、ユ

10

20

30

40

50

ーザは次のようにタイプします。

w o s i y i g e z h o n g g u o r e n (誤りは「s h」と「s」の混同)

w o s h i y i g e z o n g g u o r e n (誤りは「z h」と「z」の混同)

w o s h i y g e z h o n g g u o r e n (誤りは、「y」の後の「i」の脱落)

w o s h i y i g e z h o n g g o u r e n (誤りは、「o u」の並列)

w o s h i y i g e z h o n g g u i r e n (誤りは「i」と「o」の混同)

【0018】

発明者は、中国語などの難しい外国語でスペルミスの修正を可能にする文書処理システムと方法をすでに開発しており、自動言語認識機能により複数言語のモードレス入力が可能になっている。

【0019】

(発明の概要)

言語入力アーキテクチャにより、表音テキスト(例えば、中国語のピンイン)の入力文字列を言語テキスト(例えば、漢字)の出力文字列に変換するが、表音テキストから言語テキストに変換する際に発生するタイプミスおよび変換誤りを極力減らす。言語入力アーキテクチャは、文書処理プログラム、電子メールプログラム、表計算ソフト、ブラウザなどさまざまな分野で実装することができる。

【0020】

一実装では、言語入力アーキテクチャは入力文字列、記号、またはその他のテキスト要素を受け取るユーザインタフェースを備える。入力文字列は、表音テキストおよび非表音テキスト、さらに1つまたは複数の言語を含むことができる。このユーザインタフェースにより、ユーザは、異なるテキスト形式または異なる言語の入力のモードを切り替えることなく単一の編集行に入力テキストの文字列を入力することができる。この方法により言語入力アーキテクチャは、ユーザが使いやすいように複数の言語のモードレス入力を実現している。

【0021】

この言語入力アーキテクチャはさらに、検索エンジン、1つまたは複数のタイピングモデル、言語モデル、および異なる言語用の1つまたは複数の用語集を備える。検索エンジンは、ユーザインタフェースから入力文字列を受け取り、入力文字列を1つまたは複数のタイピングモデルに分配する。各タイピングモデルは、各候補文字列が入力文字列として間違っ

【0022】

て入力された場合のタイピング誤り確率に基づいて入力文字列に置換できる有望なタイピング候補のリストを生成するように構成されている。有望なタイピング候補は、データベースに格納できる。

【0023】

一実施形態では、タイピングモデルは、入力テキストの文字列を読み取り、音節に対応するタイプされた各文字列の文字に対応付けることによりトレーニングできる。タイプされた各文字が音節の1つに対応づけられる回数を表す頻度カウントを保持し、各音節のタイピングの確率をその頻度カウントから計算する

タイピングモデルは、入力文字列内に存在する可能性のあるタイプミスの原因となりうるタイピング候補の集まりを返す。タイピング候補は、入力文字列と同じ言語またはテキスト形式で書かれる。

【0024】

検索エンジンはタイピング候補を言語モデルに渡し、これが各タイピング候補の可能性

10

20

30

40

50

のある変換文字列となる。より具体的には、言語モデルは三重文字言語モデルであり、有望な変換出力文字列が前の2つのテキストエレメントに基づいて候補文字列を表す言語テキスト確率を求めようとする。変換文字列は、入力文字列と異なる言語または異なるテキスト形式で書かれている。例えば、入力文字列は、中国語ピンインまたはその他の表音テキストで構成され、出力文字列は漢字またはその他の言語テキストで構成される。

【0025】

タイピングおよび言語モデルで求めた確率に基づき、検索エンジンは最高の確率を示す関連するタイピング候補および変換候補を選択する。検索エンジンは、入力文字列（例えば、表音テキストで書かれている）を言語モデルから返された変換候補からなる出力文字列に変換し、入力されたテキスト形式（例えば、表音テキスト）を他のテキスト形式（例

10

【0026】

複数言語を使用する場合、出力文字列は変換候補と入力文字列の一部分（変換なし）との組み合わせを持つことができる。後者の例では、中国語ベースの言語入力アーキテクチャは、両方の変換されたピンイン - 漢字テキストを変換されていない英語テキストとともに出力する。

【0027】

ユーザインタフェースは、入力文字列の入力にそのまま使用する同じ編集行に出力文字列を表示する。この方法では、変換は自動的に行われ、ユーザが追加テキストを入力するの

20

【0028】

図全体を通して類似の構成要素および特徴を参照するのに同じ番号を使用している。

【0029】

（好ましい実施形態の詳細な説明）

本発明は、言語のある形式（例えば、表音バージョン）から言語の別の形式（例えば、書き言葉バージョン）に変換する言語入力システムおよび方法に関連する。このシステムおよび方法は、テキスト入力時に発生するスペルミスおよびタイプミスおよびある言語形式から別の言語形式への変換時に発生する変換誤りに対する誤り耐性がある。説明のため、本発明は、汎用コンピュータで実行される文書処理プログラムの一般的なコンテキスト（context）において説明される。ただし、本発明は、文書処理以外の異なる多くの環境に実装することができ、またさまざまな種類のデバイスで実施することができる。他のコンテキストとしては、電子メールプログラム、表計算ソフト、ブラウザなどを含む

30

【0030】

言語入力システムは、統計的言語モデルを採用して非常に高い精度を達成している。一実施例では、言語入力アーキテクチャは、最高確率ベースの方法による統計的言語モデリング（statistical language modeling）を使用し、自動で、単語をセグメント化し、語彙を選択し、トレーニングデータをフィルタ処理し、可能な最良の変換候補を求める。

40

【0031】

センテンスベースの統計的言語モデリング（Statistical sentence-based language modeling）では、ユーザの入力が完全であると仮定している。実際には、ユーザの入力にはタイピング誤りやスペルミスが多数ある。従って、言語入力アーキテクチャは、確率論的スペリングモデルを使用して、ありがちなタイピング誤りやスペルミスを許容しながら正しいタイピングを受け入れる1つまたは複数のタイピングモデルを含む。タイピングモデルを英語や中国語など複数言語についてトレーニングし、どれくらいの確からしさで入力シーケンスがある言語の単語であって別の言語の単語ではないかを識別するようにできる。両方のモデルは並列実行でき、その言語モデル（例えば、中国語モデル）により誘導されて最も可能性の高い文字シーケンス（

50

つまり、英語および中国語の文字)を出力する。

【0032】

<コンピュータシステム実施例>

図1は、中央処理装置(CPU)102、メモリ104、および入出力(I/O)インタフェース106を備えるコンピュータシステム実施例100を示している。CPU 102は、メモリ104およびI/Oインタフェース106と通信する。メモリ104は、揮発性メモリ(例えば、RAM)および不揮発性メモリ(例えば、ROM、ハードディスクなど)を表す。

【0033】

コンピュータシステム100は、I/Oインタフェース106を介して接続された1つまたは複数の周辺装置を備える。例として、周辺装置は、マウス110、キーボード112(例えば、英数字QWERTYキーボード、表音キーボードなど)、ディスプレイモニタ114、プリンタ116、周辺記憶装置118、およびマイクロホン120を備える。例えば、コンピュータシステムは、汎用コンピュータで実装できる。従って、コンピュータシステム100は、メモリ104に格納され、CPU 102で実行されるコンピュータのオペレーティングシステム(図に示されていない)を実装する。オペレーティングシステムは、ウィンドウ操作環境をサポートするマルチタスクオペレーティングシステムであるのが好ましい。適当なオペレーティングシステムの例として、Microsoft Corporation社のWindows(登録商標)ブランドのオペレーティングシステムがある。

【0034】

ハンドヘルドデバイス、マルチプロセッサシステム、マイクロプロセッサベースのまたはプログラム可能な家電製品、ネットワークPC、ミニコンピュータ、メインフレームコンピュータなど、他のコンピュータシステム構成を使用できることに注意されたい。さらに、図1にはスタンドアローンのコンピュータが示されているが、通信ネットワーク(例えば、LAN、インターネットなど)を介してリンクされているリモート処理デバイスによってタスクが実行される分散コンピューティング環境で言語入力システムを実用することもできる。分散コンピューティング環境では、プログラムモジュールをローカルとリモートの両方のメモリ記憶デバイスに配置できる。

【0035】

データまたは文書処理プログラム130は、メモリ104に格納され、CPU 102によって実行される。他のプログラム、データ、ファイルなども、メモリ104に格納できるが、説明を簡単にするため示していない。文書処理プログラム130は、表音テキストを受け取り、自動的に言語テキストに変換するように構成されている。より具体的には、文書処理プログラム130は、説明のためメモリ内に格納されプロセッサによって実行されるコンピュータソフトウェアとして実装されている言語入力アーキテクチャ131を実装する。文書処理プログラム130は、アーキテクチャ131に加えて他のコンポーネントも備えることができるが、そのようなコンポーネントは、文書処理プログラムに標準のものであると考えられるため、図に詳細に示したり、詳細に説明していない。

【0036】

文書処理プログラム130の言語入力アーキテクチャ131は、ユーザインタフェース(UI)132、検索エンジン134、1つまたは複数のタイピングモデル135、言語モデル136、および異なる言語用の1つまたは複数の用語集137を備える。アーキテクチャ131は、言語独立である。UI 132および検索エンジン134は、汎用であり、どの言語でも使用できる。アーキテクチャ131は、言語モデル136、タイピングモデル135、用語集137を変更することにより特定の言語に合わせる。

【0037】

検索エンジン134および言語モジュール136はいっしょに使用することで、表音テキスト-言語テキスト間のコンバータ138を形成する。タイピングモデル135の助けを借りて、コンバータ138はユーザのタイピング誤りおよびスペルミスに対し許容性を

10

20

30

40

50

持つ。この開示の目的のために、「テキスト」は、1つまたは複数の文字および/または文字以外の記号とする。「表音テキスト」は一般に、所与の言語を話すときに生じる音声を表す英数字テキストである。「言語テキスト」は、書き言葉を表す文字と非文字記号である。「非表音テキスト」は、所与の言語を話すときに生じる音声を表さない英数字テキストである。非表音テキストは、言語テキスト以外の書き言葉を表す句読点、特殊文字、および英数字テキストを含む場合がある。

【0038】

たぶん、より一般的に述べると、表音テキストは、書いたときに欧文文字ベースの文字セットを使用しない所与の言語を話したときに出る音声を表す欧文文字セット（例えば、英語のアルファベット）で表された英数字でよい。言語テキストは、所与の言語に対応する書かれた記号である。

10

【0039】

説明の目的のために、ワードプロセッサ130は、中国語ベースのワードプロセッサでのコンテキストにおいて説明され、言語入力アーキテクチャ131はピンインを漢字に変換するように構成されている。つまり、表音テキストはピンインであり、言語テキストは漢字である。しかし、言語入力アーキテクチャは、言語と無関係であり、他の言語にも使用できる。例えば、表音テキストは日本語の話し言葉の形態でよいが、言語テキストは漢字などの日本語の書き言葉を表す。他の多くの例としては、アラビア語、韓国語、インド語、その他のアジア言語などを含むが、これに限定されるものではない。

20

【0040】

表音テキストは、マウス110、キーボード112、またはマイクロホン120などの1つまたは複数の周辺入力デバイスを介して入力する。この方法で、ユーザは、キー入力または音声による表音テキスト入力が可能である。音声入力の場合、コンピュータシステムはさらに、話し言葉を受け取る音声認識モジュール（図に示されていない）を実装し、表音テキストに変換することができる。以下の説明では、キーボード112によるテキストの入力をフルサイズの標準英数字QWERTYキーボードで実行すると想定している。

【0041】

UI 132では、表音テキストを入力と同時に表示する。このUIは、グラフィカルユーザインタフェースであるのが好ましい。

【0042】

ユーザインタフェース132は、表音テキスト(P)を検索エンジン134に渡し、さらに、これは表音テキストをタイピングモデル135に渡す。タイピングモデル135は、表音テキストに誤りが含まれていると思われる場合にユーザが意図した表音テキストの適当な編集結果と考えられるさまざまなタイピング候補(TC_1, \dots, TC_N)を生成する。タイピングモデル135は、妥当な確率が設定されている複数のタイピング候補を検索エンジン134に渡し、さらに、これはタイピング候補を言語モデル136に渡す。この言語モデル136は、センテンス処理中のテキスト文字列（すなわち、context: コンテキスト（文脈））内でタイピング候補を求め、ユーザが意図した表音テキストの変換された形式を表すと考えられる言語テキストで書かれているさまざまな変換候補（ CC_1, \dots, CC_N ）を生成する。変換候補は、タイピング候補と関連付けられている。

40

【0043】

表音テキストから言語テキストへの変換は1対1変換ではない。同じあるいは類似した表音テキストが言語テキスト内の多数の文字または記号を表すことがある。従って、表音テキストの文脈は、言語テキストへの変換前に解釈される。他方、非表音テキストの変換は、通常、表示される英数字テキストが英数字入力と同じである直接的な1対1変換となる。

【0044】

変換候補(CC_1, \dots, CC_N)が検索エンジン134に戻され、このエンジンにより、タイピング候補および変換候補のうちどれがユーザが意図するものである確率が最も高

50

いかを判別する統計分析が実行される。確率が計算されると、検索エンジン 134 により、確率が最も高い候補が選択され、変換候補の言語テキストが UI 132 に返される。UI 132 では、表音テキストを変換候補の言語テキストに置き換え、同じ行に表示する。一方で、新規入力した表音テキストが新規挿入された言語テキストの前の行に表示され続ける。

【0045】

ユーザが言語テキストを検索エンジン 134 で選択したものから変更したい場合、ユーザインタフェース 132 に、選択が実際に意図した回答である可能性の高さの順序でランク付けられた他の高確率の候補の第 1 のリストが表示される。ユーザがまだ可能な候補に満足しない場合、UI 132 は可能なすべての選択肢を与える第 2 のリストを表示する。第 2 のリストは、確率またはその他の測定基準（例えば、ストロークカウントまたは中国語文字の複雑さ）に関してランク付けることができる。

10

【0046】

<言語入力アーキテクチャ>

図 2 は、言語入力アーキテクチャ 131 の詳細を示している。アーキテクチャ 131 は、タイプミスおよび変換誤りの両方を含む、言語入力の誤り耐性をサポートする。UI 132、検索エンジン 134、言語モデル 136、およびタイピングモデル 135 に加えて、アーキテクチャ 131 はさらに、エディタ 204 およびセンテンス文脈モデル 216 を備える。センテンス文脈モデル 216 は、検索エンジン 134 に結合されている。

【0047】

20

ユーザインタフェース 132 は、表音テキスト（例えば、中国語ピンインテキスト）および非表音テキスト（例えば、英語）などの入力テキストを 1 つまたは複数の周辺装置（例えば、キーボード、マウス、マイクロホン）から受け取り、その入力テキストをエディタ 204 に渡す。エディタ 204 は、検索エンジン 132 がタイピングモデル 135 および言語モデル 136 とともに入力テキストを、言語テキスト（例えば、漢字テキスト）などの出力テキストに変換するよう要求する。エディタ 204 は、出力テキストを UI 132 に戻して表示させる。

【0048】

検索エンジン 134 は、ユーザインタフェース 132 から入力テキストの文字列を受け取ると、その入力テキストの文字列をタイピングモデル 135 の 1 つまたは複数に送り、さらにセンテンス文脈モデル 216 にも送る。タイピングモデル 135 は、入力テキスト内のタイピング誤りのアприオリな確率を測定する。タイピングモデル 135 は、入力誤り（例えば、タイプミス）を修正することを実際に求めるユーザによって入力された、入力テキストの有望なタイピング候補を生成して出力する。一実施形態では、タイピングモデル 135 は候補データベース 210 内で潜在的な候補を検索する。他の実装では、タイピングモデル 135 は統計ベースのモデリングを使用して、入力テキストの有望な候補を生成する。

30

【0049】

センテンス文脈モデル 216 は、オプションで、タイピングモデル 135 で使用する検索エンジン 132 にセンテンス内のすでに入力されているテキストを送ることができる。この方法では、タイピングモデルは、テキストの新しい文字列とセンテンスにすでに入力されているテキストの文字列との組み合わせに基づいて有望なタイピング候補を生成することができる。

40

【0050】

「タイピング誤り」、「タイプミス」、および「スペルミス」という用語は、相互に入れ替えることができ、入力テキストのキー入力時に生じた誤りを指すことは明白である。音声入力の場合、このような誤りは、音声入力の不適切な認識から生じることがある。

【0051】

タイピングモデル 135 は、有望なタイピング候補をすべて返すか、または確率の低い有望なタイピング候補を取り除き、それにより、高い確率の有望なタイピング候補のみを

50

検索エンジン 134 に返すことができる。さらに、タイピングモデル 135 よりむしろ検索エンジン 134 が取り除き機能を実行できることも明白であろう。

【0052】

本発明の一態様によれば、タイピングモデル 135 は、ありがちなタイプミスを観察するため数百あるいは数千のトレーナにセンテンス入力を依頼して集めた実際のデータ 212 を使用してトレーニングされる。以下では「タイピングモデルのトレーニング」という見出しのもとでタイピングモデルおよびトレーニングについて詳述する。

【0053】

検索エンジン 134 は、タイピングモデル 135 から返された有望なタイピング候補のリストを言語モデル 136 に送る。簡単にいうと、言語モデルは、語句やセンテンスなどの所与の文脈内に、単語またはテキストの文字列がある確率を測定する。つまり、言語モデルは、項目（単語、文字、英字など）のシーケンスを取り、そのシーケンスの確率を推定することができる。言語モデル 136 は、検索エンジン 134 から有望なタイピング候補を前のテキストと組み合わせ、そのタイピング候補に対応する言語テキストの 1 つまたは複数の候補を生成する。

【0054】

コーパスデータまたはその他の種類のデータ 214 を使用して、三重文字言語モデル 136 をトレーニングする。トレーニングコーパス 214 は、新聞記事などの日々のテキストなどの一般的な任意の種類のデータ、あるいは特定の分野（例えば、医薬品）を対象とするテキストなどの環境固有のデータとすることができる。言語モデル 136 のトレーニングは、文書処理技術の分野では知られており、ここでは詳述しない。

【0055】

言語入力アーキテクチャ 131 は、入力テキストの文字列の入力時に生じる誤りを許容し、入力の文字列となる単語およびセンテンスで最も確率の高いものを返そうとする。言語モデル 136 は、タイピングモデル 135 でユーザが入力した入力文字列に対しどのセンテンスが最も妥当かを判別する際に役立つ。2つのモデルは、辞書から、入力された文字列 s が認識可能でかつ有効な単語 w である確率 $P(w | s)$ として統計的に記述することができる。ベイズの公式を使用すると、確率 $P(w | s)$ は次のように記述される。

【0056】

【数 1】

$$P(w|s) = \frac{P(s|w) \cdot P(w)}{P(s)}$$

【0057】

分母 $P(s)$ は、所与の単語 (w) を比較可能とする目的で、入力文字列 (s) が与えられる。したがって、この数式の解析は、分子の積 $P(s | w) \cdot P(w)$ のみに関係する。ここで、確率 $P(s | w)$ は、スペル、すなわちタイピングモデルの確率を表し、確率 $P(w)$ は言語モデルの確率を表す。より具体的には、タイピングモデルの確率 $P(s | w)$ は、X を入力するつもりの人が代わりに Y を入力する可能性を示す。言語モデルの確率 $P(w)$ は、与えられたセンテンスの文脈において特定の単語 (w) が生成されることとなる可能性を示す。

【0058】

ピンインを漢字に変換するコンテキストにおいては、確率 $P(w | s)$ は確率 $P(H | p)$ と言い換えることができ、 H は漢字文字列、 p はピンイン文字列を表す。目標は、確率 $P(H | p)$ を最大にする、最も確率の高い中国語文字 H を見つけることである。従って、確率 $P(H | p)$ は、入力されたピンイン文字列 p が有効な漢字文字列 H である確率である。 p は固定されており、従って、確率 $P(p)$ は与えられたピンイン文字列に対し一定であるため、ベイズ公式により、次のように確率 $P(H | p)$ が小さくなる。

【0059】

$$\begin{aligned} H &= \arg \max_{H} P(H | p) \\ &= \arg \max_{H} P(p | H) * P(H) \end{aligned}$$

【0060】

確率 $P(p | H)$ はスペルまたはタイピングモデルの確率を表す。通常、漢字文字列 H は、さらに、複数の単語 $W_1, W_2, W_3, \dots, W_M$ に分割され、確率 $P(p | H)$ は次のように推定できる。

【0061】

$$Pr(p | H) = \prod_i P(p_{f(i)} | W_i)$$

【0062】

$P_{f(i)}$ は、単語 W_i に対応するピンイン文字列のシーケンスである。

10

【0063】

従来技術の統計ベースのピンイン - 漢字変換システムでは、確率 $P(p_{f(i)} | W_i)$ は、 $P_{f(i)}$ が単語 W_i の受け入れ可能なスペルである場合に 1 に設定され、 $p_{f(i)}$ が単語 W_i の受け入れ可能なスペルでない場合に 0 に設定される。そのため、従来のシステムは、誤って入力された文字に対する耐性がない。一部のシステムでは、「南部訛りの発音」機能があり、このような問題に対応しているが、これはプリセット値確率 1 および 0 を採用している。さらに、このようなシステムは、(実際のタイピング誤りから学習する) データ駆動方式でないため、タイピング誤りのうちごくわずかししか扱えない。

【0064】

対照的に、本発明で説明している言語入力アーキテクチャでは、タイピングモデルと言語モデルの両方を利用して変換を実行する。タイピングモデルでは、実際のコーパスから確率 $P(p_{f(i)} | W_i)$ をトレーニングすることにより誤って入力した文字に対する誤り耐性が可能になっている。タイピングモデルを構築する方法は多数ある。理論的には、すべての可能な $P(p_{f(i)} | W_i)$ をトレーニングできるが、実際には、パラメータが多すぎる。トレーニングする必要のあるパラメータの個数を減らす 1 つの方法として、1 文字単語のみを考察し、発音が等価なすべての文字を単一の音節に対応づける方法がある。中国語にはおよそ 406 個の音節があり、これは本質的に $P(\text{ピンインテキスト} | \text{音節})$ をトレーニングし、各文字を対応する音節に対応づけることである。以下では「タイピングモデルのトレーニング」という見出しのもとでこれについて詳述する。

20

【0065】

言語入力アーキテクチャ 131 では、広範な確率が計算される。ピンイン - 漢字変換の目標は、確率 $P(p | H)$ を最大にする漢字文字列 H を見つけることである。これは、最大の確率を最良の漢字シーケンスとして求める W_i を選択することにより実行する。実際、よく知られている Viterbi Beam 検索のような効率的な検索方法を使用できる。Viterbi Beam 検索法の詳細については、「Automatic Speech Recognition」(Kluwer Academic Publishers、1989) という表題の Kai-Fu Lee の記事、および「Automatic Speech and Speaker Recognition - Advanced Topics」(Kluwer Academic Publishers、1996) という表題の Chin-Hui Lee、Frank K. Soong、Kuldip K. Paliwal の記事に記載されている。

30

40

【0066】

確率 $P(H)$ は、言語モデルを表し、所与の単語列のアプリオリな確率を測定する。統計的言語モデル構築の一般的な方法として、プレフィックスツリー風のデータ構造を利用して、知られているテキストのトレーニングセットから N 重文字言語モデルを構築する方法がある。広く使用されている統計的言語モデルの一実施例として、 N 重文字マルコフモデルがあり、これについては、Frederick Jelinek 著「Statistical Methods for Speech Recognition」(The MIT Press, Cambridge, Massachusetts、1997) に説明がある。プレフィックスツリーデータ構造 (a.k.a. サフィックスツリー、ま

50

たはPATツリー)の使用により、高レベルアプリケーションで言語モデルを素早く実行し、実質的にリアルタイムに実行する特性を持つ。N重文字言語モデルでは、テキスト全体を通して文字列(サイズN)内の特定のアイテム(単語、文字など)の出現数をカウントする。このカウントを使用して、そのアイテムの列の使用の確率を計算する。

【0067】

言語モデル136は、三重文字言語モデル(つまり、 $N = 3$ とするN重文字)であるのが好ましいが、状況によっては二重文字が適している場合がある。三重文字言語モデルは、英語に適しており、また中国にも十分機能するが、大きなトレーニングコーパスを利用すると想定している。

【0068】

三重文字モデルでは、次のように、次の文字を予測するためにテキスト文字列内の最も前の2つの文字を考慮する。

(a)文字(C)は、定義済み用語集を使用して離散言語テキストまたは単語(W)にセグメント化され、ツリー内の各Wは1つまたは複数のCに対応づけられる。

(b)前の2つの単語から単語のシーケンス(W_1, W_2, \dots, W_M)の確率を予測する。

【0069】

$$P(W_1, W_2, W_3, \dots, W_M) | _ P(W_n \sim W_{n-1}, W_{n-2}) \quad (1)$$

【0070】

ただし、 $P()$ は言語テキストの確率を表す。

【0071】

W_n は、現在の単語である

W_{n-1} は、前の単語である

W_{n-2} は、 W_{n-1} の前の単語である

【0072】

図3は、ユーザによって入力された入力テキスト300の例を示しており、タイピングモデル135および言語モデル136に渡される。入力テキスト300を受け取ると、タイピングモデル135は入力テキスト300を種々の方法でセグメント化し、キーボード入力時に生じる可能性のあるタイプミスを考慮した有望なタイピング候補のリストを生成する。タイピング候補302は、前の単語の終了時刻が現在の単語の開始時刻となるように、各時間フレーム内に異なる複数のセグメンテーション(segmentations)が存在する。例えば、候補302の上行は、入力テキスト300の文字列「mafangnityyis...」を「ma」、「fan」、「ni」、「try」、「yi」などにセグメント分割する。タイピング候補302の第2行は、入力文字列「mafangnityyis...」を異なる形で「ma」、「fang」、「nit」、「yu」、「xia」などにセグメント分割する。

【0073】

これらの候補は、データベースまたはその他の何らかのアクセス可能なメモリに格納できる。図3は、単なる一例にすぎず、入力テキストに対する有望なタイピング候補は多数あり得ることは明白であろう。

【0074】

言語モデル136は、センテンスの文脈で有望なタイピング候補302の各セグメントを評価し、関連する言語テキストを生成する。説明のため、有望なタイピングテキスト302の各セグメントおよび対応する有望な言語テキストはボックスにまとめられている。

【0075】

これらの候補から、検索エンジン134は、候補のうちどれがユーザが意図するものである確率が最も高いかを判別する統計分析を実行する。タイピング候補は、各行毎に互いの関連性はなく、検索エンジンはいずれかの行から種々のセグメントを自由に選択し、受け入れ可能な変換候補を定義することができる。図3の例では、検索エンジンは、ハイラ

10

20

30

40

50

イト表示になっているタイピング候補 304、306、308、310、312、および 314 が最高の確率であることを示していると判断している。これらの候補は、左から右に連結され、候補 304 の後に候補 306 が続くなどして、入力テキスト 300 の受け入れ可能な解釈を形成することができる。

【0076】

確率を計算した後、検索エンジン 134 は、確率が最高の候補を選択する。検索エンジンは次に、入力された表音テキストを選択した候補と関連する言語テキストに変換する。例えば、検索エンジンは入力テキスト 300 をボックス 304、306、308、310、312、および 314 で示されている言語テキストに変換し、エディタ 204 を介して言語テキストをユーザインタフェース 132 に返す。句読点がユーザインタフェースに届いた後、つまり新しい入力テキストの文字列が新しいセンテンス内に入ると、タイピングモデル 135 は新しいセンテンス内の新しいテキストの文字列に対する操作を開始する。

【0077】

<一般的な変換>

図 4 は、表音テキスト（例えば、ピンイン）を言語テキスト（例えば、漢字）に変換する一般的プロセス 400 を示している。このプロセスは、言語入力アーキテクチャ 131 によって実装されており、図 2 をさらに参照して説明する。

【0078】

ステップ 402 では、ユーザインタフェース 132 は、ユーザが入力したピンインなどの表音テキスト列を受け取る。入力テキストの文字列には、1 つまたは複数のタイプミスが含まれる。UI 132 は、エディタ 204 を介して入力テキストを検索エンジン 134 に渡し、検索エンジンは入力テキストをタイピングモデル 135 とセンテンス文脈モデル 216 に配送する。

【0079】

ステップ 404 では、タイピングモデル 135 は入力テキストに基づいて有望なタイピング候補を生成する。候補を導く一方法として、入力テキストの文字列を異なるパーティションに分割し、その入力文字列のセグメントに最もよく類似する候補をデータベース内で検索する。例えば、図 3 で、候補 302 は可能なセグメント「ma」、「fan」などを示すセグメンテーションを持つ。

【0080】

有望なタイピング候補が、検索エンジン 134 に返され、その後、言語モデル 136 に伝達される。言語モデル 136 は、有望なタイピング候補と以前の入力テキストとを組み合わせ、タイピング候補に対応する言語テキストの 1 つまたは複数の候補を生成する。例えば、図 3 の候補 302 を参照すると、言語モデルはボックス 302 a - j 内に言語テキストを可能な出力テキストとして返す。

【0081】

ステップ 406 では、検索エンジン 134 は、候補のうちどれがユーザが意図するものである確率が最も高いかを判別する統計分析を実行する。表音テキストに最も可能性の高いタイピング候補を選択した後、検索エンジンは入力された表音テキストをタイピング候補と関連する言語テキストに変換する。この方法により、表音テキストの入力時のユーザによる入力誤りがなくなる。検索エンジン 134 は、エディタ 204 を介して誤りのない言語テキストを UI 132 に返す。ステップ 408 では、変換された言語テキストは、ユーザが表音テキストを入力し続けている UI 132 の画面上の同じ行内位置に表示される。

【0082】

<タイピングモデルのトレーニング>

上で指摘したように、タイピングモデル 135 は、確率 $P(s|w)$ に基づいて処理がなされる。タイピングモデルでは、入力テキストを出力テキストに変換するのに使用できる異なるタイピング候補の確率を計算し、有望な候補を選択する。この方法で、タイピングモデルは、タイピング誤りが存在していても入力テキストの有望なタイピング候補を返

10

20

30

40

50

すことにより誤りを許容する。

【0083】

本発明の一態様は、実際のデータからのタイピングモデル $P(s|w)$ のトレーニングに関するものである。タイピングモデルは、数百または好ましくは数千などできる限り多くのトレーナによるテキスト入力に基づいて開発またはトレーニングされる。トレーナは同じまたは異なるトレーニングデータを入力し、入力されたデータとトレーニングデータとの差異をタイピング誤りとして捕らえる。目標は、同じトレーニングテキストをタイプさせ、タイピングでの誤りの個数またはタイピング候補に基づいて確率を求めることである。このようにして、タイピングモデルはトレーナのタイピング誤りの確率を学習する。

【0084】

図5は、プロセッサ502、揮発性メモリ504、および不揮発性メモリ506を備えるトレーニング用コンピュータ500を示している。トレーニング用コンピュータ500では、ユーザが入力したデータ510から確率512（つまり、 $P(s|w)$ ）を求めるトレーニングプログラム508を実行する。トレーニングプログラム508は、プロセッサ502で実行するように図に示されているが、不揮発性メモリ506のストレージからプロセッサにロードされる。トレーニング用コンピュータ500は、オンザフライでの入力時に、あるいは収集しメモリに格納した後にデータ510に基づいてトレーニングを行うように構成することができる。

【0085】

説明のため、中国語用に手直ししたタイピングモデルを考察し、中国語ピンインテキストは中国語文字テキストに変換する。この場合、数千人の人々にピンインテキストを入力してくれるよう勧誘する。好ましくは、数千個またはそれ以上のセンテンスを各人から収集し、目標はタイピングでの誤りの種類および個数が類似するようにすることである。タイピングモデルは、検索エンジンからピンインテキストを受け取り、入力文字列内の文字の置き換えに使用できる有望な候補を供給するように構成されている。

【0086】

タイピングモデル135をトレーニングするためにさまざまな手法を使用できる。一方では、タイピングモデルは、単一文字テキストについて、すべての同等な発音の文字テキストを単一音節に対応付けることにより直接トレーニングされる。例えば、中国語ピンインには400を超える音節がある。音節を与える表音テキストの確率（例えば、 $P(\text{ピンインテキスト}|\text{音節})$ ）をトレーニングし、各文字テキストを対応する音節に対応付ける。

【0087】

図6は、音節対応付けトレーニング手法600を示している。ステップ602で、トレーニングプログラム508は、トレーナが入力したテキスト文字列を読みとる。テキスト文字列は、センテンスでも、また単語および/または文字のその他のグループでもよい。プログラム508は、音節をテキストの文字列内の対応する英字に合わせるか、または対応付ける（ステップ604）。各テキスト文字列では、各音節に対応付けた英字の頻度が更新される（ステップ606）。これは、ステップ608から「はい」分岐で表されているように、トレーナにより入力されたトレーニングデータに含まれるテキスト文字列ごとに繰り返される。最終的に、入力されたテキスト文字列は、中国語ピンインの多くのまたはすべての音節を表す。ステップ608から「いいえ」分岐により表されるように、すべての文字列が読み込まれたら、トレーニングプログラムは、ユーザが各音節をタイピングする確率 $P(\text{ピンインテキスト}|\text{音節})$ を決定する（ステップ610）。一実装では、そのタイピングする確率 $P(\text{ピンインテキスト}|\text{音節})$ は、すべての音節を最初に正規化して決定される。

【0088】

各音節は、隠しマルコフモデル（HMM）として表すことができる。各入力キーは、HMMで対応付けられている状態のシーケンスとして表示できる。正しい入力および実際の入力をすりあわせて状態間の遷移確率を求める。異なるHMMを使用して、異なる技能レ

10

20

30

40

50

ベルのタイピストをモデル化することができる。

【0089】

中国語で406個すべての音節をトレーニングするには、大量のデータが必要である。このデータ要件を緩和するために、異なる音節内の同じ文字を1つの状態として結びつける。これにより、状態の個数は27にまで減らされる（つまり、「a」から「z」までの26個の異なる英字に、不明な文字を表す文字1つ）。このモデルは、三重文字言語モデルを利用するViterbi beam検索法に統合することもできる。

【0090】

さらに他のトレーニング手法では、トレーニングは英字の挿入（ x ）、

【0092】

英字の削除（x ）、

【0094】

および一方の文字を他方に置換（x y ）

【0096】

）などの単一文字編集の確率に基づく。このような単一文字編集の確率は次のように統計的に表すことができる。

【0097】

置換：P（xをyで置換）

挿入：P（xをyの前／後に挿入）

削除：P（xをyの前／後に削除）

【0098】

各確率（P）は、本質的に二重文字タイピングモデルであるが、隣接する文字を超えるかなり広い文脈のテキストを考慮したN重文字タイピングモデルに拡張することもできる。従って、入力テキストの可能な文字列について、タイピングモデルは、まず正しい文字シーケンスを供給し、次に動的プログラミングを使用して正しい文字シーケンスを与えられた文字シーケンスに変換する最低コスト経路を求めることにより、すべての可能な文字シーケンスを生成する確率を持つ。コストを、最小数の誤り文字または他の何らかの測定基準として決めることができる。実際には、この誤りモデルは、Viterbi Beam検索法の一部として実装できる。

【0099】

タイピング誤りまたはスペルミス以外の種類の誤りは、本発明の範囲内でトレーニングできることは明白であろう。さらに、異なるトレーニング手法を使用して、本発明の範囲から逸脱することなくタイピングモデルをトレーニングすることができることも明白であろう。

【0100】

<モードレス入力の多言語トレーニング>

言語入力システムを悩ます他のやっかいな問題として、2つまたはそれ以上の言語を入力したときのモードの切り替えの必要性である。例えば、中国語でタイプしているユーザは、英語の単語を入力したい場合がある。従来の入力システムでは、ユーザは英単語のタイピングと中国語の単語のタイプのモード切替が必要である。残念なことに、ユーザが切り替えを忘れやすいということである。

【0101】

言語入力アーキテクチャ131（図1）をトレーニングして混合言語入力を受け入れトレーニングすることができ、従って、多言語文書処理システムにおいて2つまたはそれ以上の言語間のモード切替をなくすことができる。これは、「モードレス入力」と呼ばれる。

【0102】

この言語入力アーキテクチャは、中国語と英語を区別するなど、異なる言語の単語を自動的に識別するスペル／タイピングモデルを実装している。これは、多くの正当な英単語は正当なピンイン文字列であるため容易ではない。さらに、ピンイン、英語、および中国

10

20

30

40

50

語文字の間にスペースが入らないため、入力時に曖昧さが増すことがある。以下のベイズ規則を使用すると、

【0103】

$$\begin{aligned} H &= \arg \max_{\underline{p}} P(H | p) \\ &= \arg \max_{\underline{p}} P(p | H) * P(H) \end{aligned}$$

【0104】

目的関数は、英語ではスペルモデル $P(\underline{p} | H)$ 、中国語では言語モデル $P(H)$ の2つの部分があることで特徴付けることができる。

【0105】

混合言語入力を取り扱う一方法として、第2言語からの単語を第1言語の特殊カテゴリとして取り扱うことにより、第1言語（例えば、中国語）の言語モデルをトレーニングする方法がある。例えば、第2言語からの単語を第1言語の単一単語として取り扱う。

10

【0106】

例えば、中国語ベースの文書処理システムでは、英語キーボードを入力デバイスとして使用する。中国語ベースの文書処理システムで採用しているタイピングモデルは、中国語言語モデルであって、英単語と中国語単語を混ぜたテキストでトレーニングされる。

【0107】

混合言語入力を取り扱う第2の方法として、言語入力アーキテクチャで2つのタイピングモデル、中国語タイピングモデルと英語タイピングモデルを実装し、それぞれを別々にトレーニングする方法がある。つまり、中国語タイピングモデルは、上で説明した方法で
トレーナにより入力された表音列などのキーボード入力のストリーム上でトレーニングされ、英語タイピングモデルは英語を話すトレーナによって入力された英語テキスト上で
トレーニングされる。

20

【0108】

英語タイピングモデルは、以下の組み合わせとして実装することができる。

1．中国語テキストに挿入された実際の英語上でトレーニングした一重文字言語モデル。このモデルは、多くの頻繁に使用される英単語を取り扱えるが、見たことのない英単語は予測できない。

【0109】

2．3音節確率の英語スペルモデル。このモデルは、すべての3音節シーケンスに対し確率が0でないが、英語に似た単語になる可能性のある単語については確率が高くなる。これは、実際の英単語からもトレーニングでき、見たことのない英単語も取り扱える。

30

【0110】

これらの英語モデルは、一般に、英語テキストに対しては非常に高い確率、英語テキストのように見える英字列には高い確率、非英語テキストには低い確率を返す。

【0111】

図7は、図2のアーキテクチャ131から修正された言語入力アーキテクチャ700を示しており、これは複数のタイピングモデル135(1)～135(N)を採用している。各タイピングモデルは、特定の言語に合わせて構成されている。各タイピングモデル135は、単語と、特定の言語に共通する誤りを使用して別々にトレーニングされる。従って、別々のトレーニングデータ212(1)～212(N)が、関連するタイピングモデル135(1)～135(N)について供給される。実施例では、英語に対して1つのタイピングモデル、中国語に対してもう1つのタイピングモデルというようにタイピングモデルを2つだけ使用している。ただし、言語入力アーキテクチャを修正して、2つよりも多いタイピングモデルを含めて、2つよりも多い言語の入力に対応するようにできることは明白であろう。また、言語入力アーキテクチャは、日本語、韓国語、フランス語、ドイツ語などの他の多くの多言語文書処理システムでも使用できることも指摘しておくべきであろう。

40

【0112】

言語入力アーキテクチャの操作時に、英語タイピングモデルは中国語タイピングモデル

50

と並列に動作する。２つのタイピングモデルは互いに競合しており、入力したテキスト文字列が中国語文字列（誤りを含む）または英語文字列（さらに潜在的に誤りを含む）である可能性のある確率を計算することにより入力テキストが英語か中国語かを識別する。

【 0 1 1 3 】

入力テキストの文字列またはシーケンスが明確に中国語ピンインテキストの場合、中国語タイピングモデルは英語タイピングモデルよりもかなり高い確率を返す。そこで、言語入力アーキテクチャは、入力されたピンインテキストを漢字テキストに変換する。入力テキストの文字列またはシーケンスが明確に英語（例えば、サーネーム、頭字語（「 I E E E 」、「 M i c r o s o f t 」、「 I N T E R N E T ）」、など）の場合、英語タイピングモデルは中国語タイピングモデルよりもかなり高い確率を示す。従って、このアーキテクチャは、英語タイピングモデルに基づいて入力テキストを英語テキストに変換する。

10

【 0 1 1 4 】

入力テキストの文字列またはシーケンスが曖昧な場合、中国語および英語タイピングモデルは、中国語か英語かわからない曖昧さを解消するためにさらなる文脈から情報が得られるまで確率を計算し続ける。入力テキストの文字列またはシーケンスが中国語にも英語にも似ていない場合、中国語タイピングモデルは英語タイピングモデルよりも許容性が低い。そのため、英語タイピングモデルは確率が、中国語タイピングモデルよりも高くなる。

【 0 1 1 5 】

20

多言語変換を説明するために、ユーザが「私は I N T E R N E T マガジンを読むのが好きだ」という意味のテキスト文字列「 w o a i d u i n t e r n e t z a z h i 」を入力すると仮定する。初期文字列「 w o a i d u 」を受け取った後、中国語タイピングモデルは、英語タイピングモデルよりも高い確率となり、入力テキストのその部分を「 I N T E R N E T 」に変換する。このアーキテクチャは、続いてタイプした曖昧な部分「 i n t e r n e 」を英字「 t 」がタイプされるまで探し続ける。このときに、英語タイピングモデルは、「 I N T E R N E T 」について中国語タイピングモデルよりも高い確率を返し、言語入力アーキテクチャは入力テキストのこの部分を「 I N T E R N E T 」に変換する。次に、中国語タイピングモデルは、「 z a z h i 」について英語タイピングモデルよりも高い確率を示し、言語入力アーキテクチャは入力テキストのその部分について変換する。

30

【 0 1 1 6 】

< 多言語入力変換 >

図 8 は、タイプミスとともに入力された多言語入力テキスト文字列を誤りのない多言語出力テキスト文字列に変換するプロセス 8 0 0 を示している。このプロセスは、言語入力アーキテクチャ 7 0 0 によって実装されており、図 7 をさらに参照して説明する。

【 0 1 1 7 】

ステップ 8 0 2 では、ユーザインタフェース 1 3 2 は、多言語入力テキスト文字列を受け取る。これは、少なくとも１つの他の言語（例えば、英語）の表音単語（例えば、ピンイン）および単語を含む。入力テキストはさらに、ユーザが表音単語および第 2 言語の単語を入力したときのタイプミスも含む場合がある。U I 1 3 2 は、エディタ 2 0 4 を介して多言語入力テキスト文字列を検索エンジン 1 3 4 に渡し、検索エンジンは入力テキストをタイピングモデル 1 3 5 (1) ~ 1 3 5 (N) とセンテンス文脈モデル 2 1 6 に分配する。

40

【 0 1 1 8 】

タイピングモデルはそれぞれ、ステップ 8 0 4 (1) ~ 8 0 4 (N) によって表されるような入力テキストに基づいて有望なタイピング候補を生成する。ステップ 8 0 6 では、妥当な確率が設定された有望なタイピング候補が検索エンジン 1 3 4 に返される。ステップ 8 0 8 では、検索エンジン 1 3 4 がタイピング確率とともにタイピング候補を言語モデル 1 3 6 に送る。ステップ 8 1 0 では、言語モデルが有望なタイピング候補と以前の入力テキストとを組み合わせることによってセンテンスベースの文脈を提供し、図 3 に示した

50

ように、複数のタイピング候補を通じてセグメント経路を選択することによりタイピング候補に対応する言語テキストの1つまたは複数の変換候補を生成する。ステップ812では、検索エンジン134は統計分析を実行して、ユーザが意図する最高の確率を示す変換候補を選択する。

【0119】

ステップ814では、テキスト文字列に対する最も有望な変換候補が出力テキスト文字列に変換される。出力テキスト文字列は、言語テキスト（例えば、漢字）と第2言語（例えば、英語）を含むが、タイピング誤りは省かれる。検索エンジン134は、エディタ204を介して誤りのない出力テキストをUI 132に返す。ステップ816では、変換された言語テキストは、ユーザが表音テキストを入力し続けているUI 132の画面上の同じ行内位置に表示される。

10

【0120】

上の例では、中国語が主言語であり、英語は第2言語である。2つの言語は両方とも、主言語として指定できることは明白であろう。さらに、2つよりも多い言語は混合入力テキスト文字列を形成することができる。

【0121】

< 結論 >

上の説明では、構造機能および/または方法論的動作に固有の言語を使用しているが、付属の請求項で定義されている本発明は説明した特定の機能または動作に限られるわけではない。むしろ、特定の機能および動作は、本発明を実装する実施例として開示されている。

20

【図面の簡単な説明】

【図1】 言語入力アーキテクチャを実装する言語固有のワードプロセッサを備えるコンピュータシステムのブロック図である。

【図2】 言語入力アーキテクチャの実施例のブロック図である。

【図3】 構文解析またはセグメント化して異なる音節群に分けたテキスト文字列および、テキスト文字列に誤りが含まれると仮定してそれらの音節を置き換えるのに使用することができる候補を説明しているブロック図である。

【図4】 言語入力アーキテクチャで実行される一般的変換を示す流れ図である。

【図5】 言語入力アーキテクチャで採用されている確率ベースのモデルをトレーニングするために使用されるトレーニングコンピュータのブロック図である。

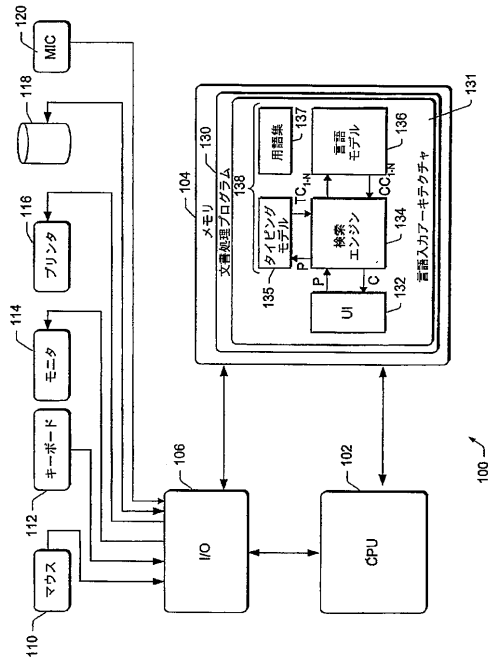
30

【図6】 ートレーニング手法を説明する流れ図である。

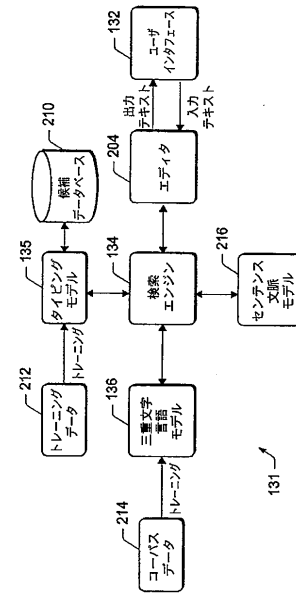
【図7】 複数のタイピングモデルを使用する、言語入力アーキテクチャの他の実施例のブロック図である。

【図8】 他言語変換プロセスを説明する流れ図である。

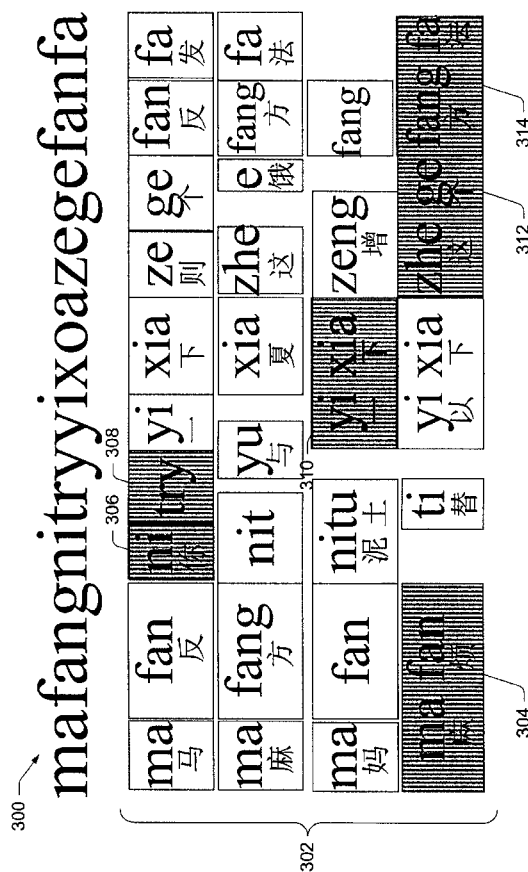
【図 1】



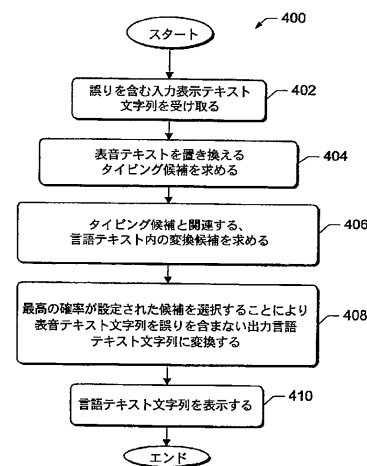
【図 2】



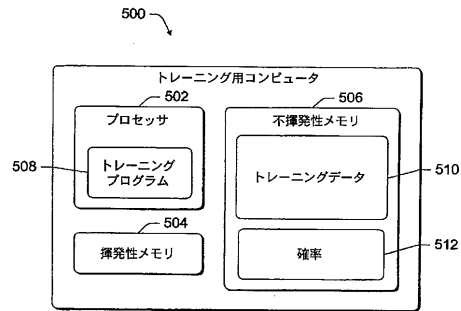
【図 3】



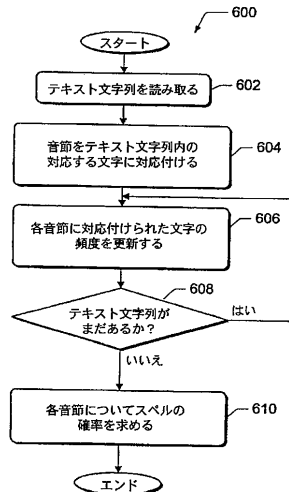
【図 4】



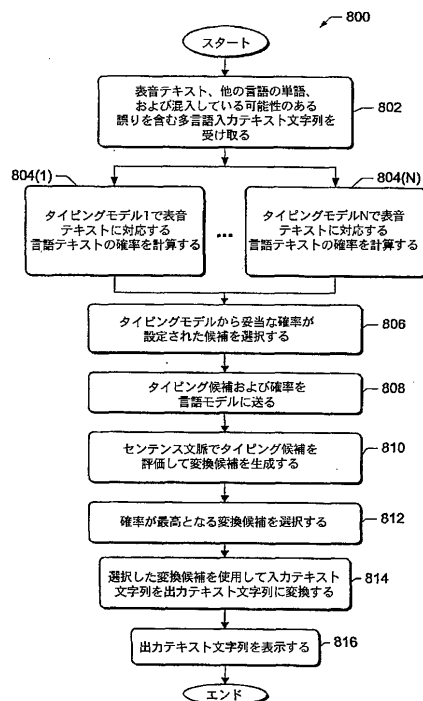
【図 5】



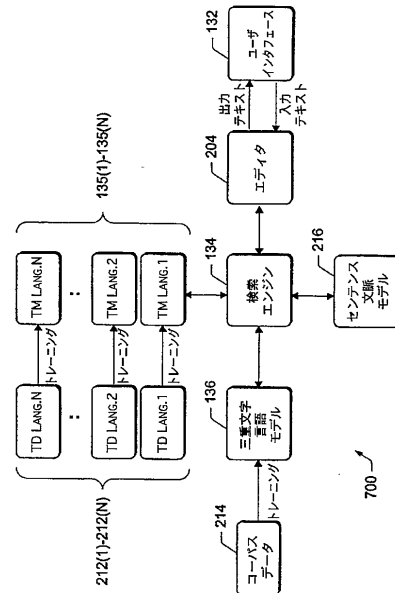
【図 6】



【図 8】



【図 7】



フロントページの続き

(72)発明者 ズエン チェン

中華人民共和国 北京 ハイジャン ディストリクト ズイチュン ロード (番地なし) ビル
ディング 9 ルーム 1801

(72)発明者 ジアン ハン

中華人民共和国 北京 ハイダン ディストリクト ドンワンズアン コミュニティー (番地なし)
ビルディング 5 ルーム 702

合議体

審判長 山崎 達也

審判官 辻本 泰隆

審判官 田中 秀人

(56)参考文献 特開平11-096141(JP,A)

特開平05-108647(JP,A)

特開平03-030048(JP,A)

特開平11-175518(JP,A)

特開平07-057055(JP,A)

特開平07-036878(JP,A)

特開平04-167051(JP,A)

特開平01-193968(JP,A)

特開昭60-097426(JP,A)

特開平11-003338(JP,A)

特開平10-232863(JP,A)

Mark Kernighan 外2名, "A Spelling Correction Program Based on a Noisy Channel Model", Proc. of the 13th Conference on Computational Linguistics, (米), Association for Computational Linguistics, 1990, Vol.2, p.205-210

匂坂 芳典, "音声処理技術とその応用 4. 音声認識技術", 情報処理, 社団法人情報処理学会, 1997年11月15日, 第38巻, 第11号, p.992-997

(58)調査した分野(Int.Cl., DB名)

G06F17/21