(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2004-158038
(P2004-158038A)
(43)公開日　平成16年6月3日(2004.6.3)

(51)Int.Cl.[7]	F I			テーマコード（参考）
G06F 17/21	G06F	17/21	550A	5B009
G06F 17/28	G06F	17/28	Z	5B091

審査請求 有　請求項の数 4　OL　（全 16 頁）

(21)出願番号　特願2004-17845(P2004-17845)
(22)出願日　平成16年1月27日(2004.1.27)
(62)分割の表示　特願2001-311329(P2001-311329)の分割
原出願日　平成13年10月9日(2001.10.9)

特許法第３０条第１項適用申請有り　平成１３年７月１０日　社団法人電子情報通信学会発行の「電子情報通信学会技術研究報告　信学情報　Ｖｏｌ．１０１　Ｎｏ．１９０」に発表

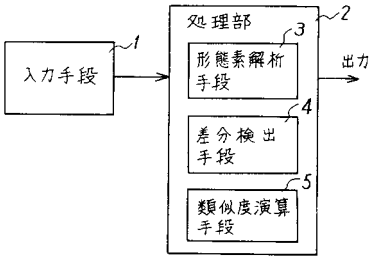
(71)出願人　301022471
独立行政法人情報通信研究機構
東京都小金井市貫井北町４－２－１
(74)代理人　100103827
弁理士　平岡　憲一
(74)代理人　100097836
弁理士　福井　國敞
(72)発明者　村田　真樹
東京都小金井市貫井北町４－２－１　独立行政法人通信総合研究所内
(72)発明者　井佐原　均
東京都小金井市貫井北町４－２－１　独立行政法人通信総合研究所内

Fターム(参考)　5B009 QA01　VB01
5B091 CA01　CA02

(54)【発明の名称】言語処理システム及びプログラム

(57)【要約】
【課題】予稿と講演データ等の第一の言語データと第二の言語データとの対応付けを簡単に行うこと。
【解決手段】対応づけの目印になる記号を含む第一の言語データとその記号を含まない第二の言語データの複数の言語データを入力する入力手段１と、前記複数の言語データの共通部分と差分部分を検出する差分検出手段４とを備え、前記差分検出手段４は、前記共通部分からはそのまま言語データを取り出し、前記差分部分からは前記第二の言語データが対応する側の差分部分から言語データを取り出し、かつ、前記第一の言語データが対応する側の差分部分からは前記記号を取り出してそれらをそのままの順番のまま並べることにより、前記記号が挿入された第二の言語データを持つことにより、前記複数の言語データを対応づける。
【選択図】　図１

言語処理システムの説明図



【特許請求の範囲】
【請求項１】
対応づけの目印になる記号を含む第一の言語データとその記号を含まない第二の言語データの複数の言語データを入力する入力手段と、
順序情報を保持したまま一致部分を最大になるように複数のデータの対応づけを行うシステムを用いて前記複数の言語データの共通部分と差分部分を検出する差分検出手段とを備え、
前記差分検出手段は、前記共通部分からはそのまま言語データを取り出し、前記差分部分からは前記第二の言語データが対応する側の差分部分から言語データを取り出し、かつ、前記第一の言語データが対応する側の差分部分からは前記記号を取り出してそれらをそのままの順番のまま並べることにより、前記記号が挿入された第二の言語データを持つことにより、前記複数の言語データを対応づけることを特徴とした言語処理システム。
【請求項２】
予稿データとその講演データを対応づけるためのシステムであって、
前記複数の言語データが、予稿データとその講演データであることを特徴とする請求項１記載の言語処理システム。
【請求項３】
前記順序情報を保持したまま一致部分を最大になるように複数のデータの対応づけを行うシステムとしてdiffコマンドを用いることを特徴とした請求項１又は２記載の言語処理システム。
【請求項４】
対応づけの目印になる記号を含む第一の言語データとその記号を含まない第二の言語データの複数の言語データを入力する処理と、
順序情報を保持したまま一致部分を最大になるように複数のデータの対応づけを行うシステムを用いて前記複数の言語データの共通部分と差分部分を検出する処理と、
前記共通部分からはそのまま言語データを取り出し、前記差分部分からは前記第二の言語データが対応する側の差分部分から言語データを取り出し、かつ、前記第一の言語データが対応する側の差分部分からは前記記号を取り出してそれらをそのままの順番のまま並べることにより、前記記号が挿入された第二の言語データを持つことにより、前記複数の言語データを対応づける処理とを、
コンピュータに実現させるためのプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、差分検出を行うdiff（ディフ）コマンドのような順序情報を保持したまま一致部分が最大になるように複数のデータの対応づけを行うシステムを用いた言語処理システム及びプログラムに関する。
【背景技術】
【０００２】
従来、例えば、日本語文と英語文の対応付け、講演と予稿の対応付け、質問文と知識データの対応付け等は、判断処理としてのプログラムが複雑なものであった。
【発明の開示】
【発明が解決しようとする課題】
【０００３】
上記従来の対訳コーパスの対応付け、従来の講演と予稿の対応付け及び質問文と知識データの対応付けを行うものは、複雑なプログラムが必要であり、簡単に対応付けを行うのが困難なものであった。
【０００４】
本発明は上記問題点の解決を図り、diffコマンドを用いて、対訳コーパスの対応付け、講演と予稿の対応付け等の複数の言語データの対応付けを簡単に行えるようにすることを

目的とする。
【課題を解決するための手段】
【０００５】
図１は本発明の言語処理システムである。図１中、１は入力手段、２は処理部、３は形態素解析手段、４は差分検出手段、５は類似度演算手段である。
【０００６】
本発明は、前記従来の課題を解決するため次のような手段を有する。
【０００７】
（１）：対応づけの目印になる記号を含む第一の言語データとその記号を含まない第二の言語データの複数の言語データを入力する入力手段１と、順序情報を保持したまま一致部分を最大になるように複数のデータの対応づけを行うシステムを用いて前記複数の言語データの共通部分と差分部分を検出する差分検出手段４とを備え、前記差分検出手段４は、前記共通部分からはそのまま言語データを取り出し、前記差分部分からは前記第二の言語データが対応する側の差分部分から言語データを取り出し、かつ、前記第一の言語データが対応する側の差分部分からは前記記号を取り出してそれらをそのままの順番のまま並べることにより、前記記号が挿入された第二の言語データを持つことにより、前記複数の言語データを対応づける。このため、例えば、対応づけの目印になる記号である章の情報だけを残して予稿のデータ（第一の言語データ）を消し去ることにより、講演データ（第二の言語データ）に章情報を挿入することができ、予稿と講演との対応付けを簡単に行うことができる。
【０００８】
（２）：前記（１）の言語処理システムにおいて、予稿データとその講演データを対応づけるためのシステムであって、前記複数の言語データが、予稿データとその講演データであるものとする。このため、対応づけの目印になる記号である章の情報だけを残して予稿のデータを消し去ることにより、講演データに章情報を挿入することができ、予稿と講演との対応付けを簡単に行うことができる。
【０００９】
（３）：前記（１）又は（２）の言語処理システムにおいて、前記順序情報を保持したまま一致部分を最大になるように複数のデータの対応づけを行うシステムとしてdiffコマンドを用いる。このため、複数の言語データの対応付けを簡単に行うことができる。
【００１０】
（４）：対応づけの目印になる記号を含む第一の言語データとその記号を含まない第二の言語データの複数の言語データを入力する処理と、順序情報を保持したまま一致部分を最大になるように複数のデータの対応づけを行うシステムを用いて前記複数の言語データの共通部分と差分部分を検出する処理と、前記共通部分からはそのまま言語データを取り出し、前記差分部分からは前記第二の言語データが対応する側の差分部分から言語データを取り出し、かつ、前記第一の言語データが対応する側の差分部分からは前記記号を取り出してそれらをそのままの順番のまま並べることにより、前記記号が挿入された第二の言語データを持つことにより、前記複数の言語データを対応づける処理とを、コンピュータに実現させるためのプログラムとする。このため、このプログラムをコンピュータにインストールすることで、予稿と講演等の複数の言語データの対応付けを容易に実現することができる言語処理システムを容易に提供することができる。
【発明の効果】
【００１１】
本発明によれば次のような効果がある。
【００１２】
（１）：差分検出手段で、共通部分からはそのまま言語データを取り出し、差分部分からは第二の言語データが対応する側の差分部分から言語データを取り出し、かつ、第一の言語データが対応する側の差分部分からは対応づけの目印になる記号を取り出してそれらをそのままの順番のまま並べることにより、前記記号が挿入された第二の言語データを持

つことにより、複数の言語データを対応づけるため、例えば、対応づけの目印になる記号である章の情報だけを残して予稿のデータ（第一の言語データ）を消し去ることにより、講演データ（第二の言語データ）に章情報を挿入することができ、予稿と講演との対応付けを簡単に行うことができる。

【００１３】

（２）：前記複数の言語データが、予稿データとその講演データであるものとするため、対応づけの目印になる記号である章の情報だけを残して予稿のデータを消し去ることにより、講演データに章情報を挿入することができ、予稿と講演との対応付けを簡単に行うことができる。

【００１４】

（３）：前記順序情報を保持したまま一致部分を最大になるように複数のデータの対応づけを行うシステムとしてdiffコマンドを用いるため、複数の言語データの対応付けを簡単に行うことができる。

【発明を実施するための最良の形態】

【００１５】

（１）：言語処理システムの説明

図１は本発明の言語処理システムの説明図である。図１において、入力手段１は、差分検出を行う言語データを入力するものである。処理部２は、入力されたデータの処理を行うものである。形態素解析手段３は、入力された言語データを辞書と文法を用いて最適な単語列に分割するものである。差分検出手段４は、対応関係のある複数の言語データをdiffコマンドを用いて差分の検出を行うものである。類似度演算手段５は、差分検出手段４を用いて質問文と知識データとの類似度を求め、類似度の大きい知識データの質問文の疑問詞に対応する部分を出力するものである。

【００１６】

（２）：diff、 mdiff、 mdiffc の説明

(1) diffの説明

diff（ディフ）とは、ＵＮＩＸ（ユニックス）（登録商標）のファイル比較ツールdiffのことである。このdiffコマンドは、与えられた二つのファイルの差分を順序情報を保持したまま行を単位として出力するものである。

【００１７】

例えば、

今日
学校へ
いく

ということが書いてあるファイル(1) と

今日
大学へ
いく

ということが書いてあるファイル(2) があるとする。これらのdiffをとると、差分が

＜学校へ
＞大学へ

のような形で出力される。

【００１８】

(2) mdiffの説明

ところで、diffコマンドには、－Ｄオプションという便利なオプションがある。このオプションを付けてdiffコマンドを使うと差分部分だけでなく共通部分も出力される。つまり、ファイルのマージが実現される。また、差分部分は、Ｃ（プログラム言語）のプリプロセッサなどで使われるifdef 文などで表現されるが、ここでは、ifdef 文は、見にくいので差分部分は以下のように表示することにする。

【００１９】

；▽▽▽▽▽▽
（一つ目のファイルにだけある部分）
；●●●
（二つ目のファイルにだけある部分）
；△△△△△△

ここでは、“；▽▽▽▽▽▽”は差分部分の始まりを、“；△△△△△△”は差分部分の終わりを意味し、“；●●●”は差分を構成する二つのデータの境界を意味する。

【００２０】

本実施の形態では、－Ｄオプションを付けて更にifdef の部分を上記のように表示（整形）して、ファイルのマージを行う場合のdiffをmdiff （エムディフ）と呼ぶ（m はmerge の mである）。

【００２１】

実際に、先ほどのデータ（ファイル(1) とファイル(2) ）に対して、mdiff をかけてみると、以下のような結果になる。

【００２２】

今日
；▽▽▽▽▽▽
学校へ
；●●●
大学へ
；△△△△△△
いく

これは「今日」が一致し、「学校へ」と「大学へ」が差分となり、「いく」がまた共通部分となっている。このように、mdiff の出力はdiffと異なり一致部分も出力されるために分かりやすい。

【００２３】

また、mdiff の結果からは元の二つのファイルのデータを完全に復元することができる。共通部分と、差分部分の黒丸（；●●●）の上側だけを取り出すと、

今日
学校へ
いく

のように一つ目のファイルの情報が取り出される。また、共通部分と、差分部分の黒丸（；●●●）の下側だけを取り出すと、

今日
大学へ
いく

のように二つ目のファイルの情報が取り出される。このように、もとの情報を完全に復元できることになる。

【００２４】

また、mdiff では、一致部分は片方のデータにあったものだけを表示し、不一致部分のみ両方のデータのものを表示するために、元の二つのデータよりもデータ量は削減できるが、上記のように元の情報を完全に復元できるために、復元できる状態でデータ量を削減するという意味ではmdiff はデータ圧縮を実現しているものといえる。

【００２５】

図２はmdiff の言語処理システムの説明図である。図２において、mdiff の言語処理システムには、ＵＮＩＸdiff処理部１１、整形部１２が設けてある。ＵＮＩＸdiff処理部１１は、入力された二つのファイルのdiffによる差分部分と共通部分を出力するものである。整形部１２は、ＵＮＩＸdiff処理部１１の出力を見やすい表現に整形するものである。

【００２６】

図３はmdiff によるフローチャートである。以下、図３の処理Ｓ１～Ｓ４に従って説明

する。
【００２７】
Ｓ１：ＵＮＩＸdiff処理部１１に、入力として二つのファイルが与えられる。
【００２８】
Ｓ２：ＵＮＩＸdiff処理部１１で、二つのファイルの一致・不一致部分を検出する（ここでのdiffはオプション－Ｄを付けておき一致部分の検出も行う）。
【００２９】
Ｓ３：整形部１２において、ＵＮＩＸdiff処理部１１の出力を整形する。具体的には“diff－Ｄ記号”の出力のifdef 文を“；▽▽▽▽▽▽”などの見やすい表現に直す処理を行う。
【００３０】
Ｓ４：整形部１２は、整形された結果を出力する。
【００３１】
3)：mdiffcの説明
【００３２】
次に、文字を単位としたmdiff を考える。言語処理の場合は、文字単位を差分で取りたい場合が多い。そのようなときは一度ファイルの中身の情報を、一文字ずつ改行をして出力したファイルでmdiff をとればよい。例えば先のファイル(1) の情報だと、
今
【００３３】
日
学
校
へ
い
く
という形にしてから、mdiff をとればよい。この一文字単位でmdiff をかけることをmdiffcと呼ぶ（mdiffcの cはcharacter のｃ）。
【００３４】
diffの表示は見にくく、mdiff はdiffで表示される情報を完全に含むので以降の説明は、mdiff を用いて行う。
【００３５】
（３）：差分検出及び書き換え規則の獲得の説明
(1) 複数システムの出力の差分検出の説明
以前、juman のシステムのバージョンが複数乱立しているとき、この複数のjuman の出力をmdiff によりマージして形態素解析結果の品質を向上させるようなことをしていた（参考文献、村田真樹，日本語文章における名詞の指示対象の推定，京都大学工学部博士論文，(1995)、石間衛，藤井敦，石川徹也，日本語形態素・構文解析システムJEMONIの開発と評価について，情報処理学会自然言語処理研究会 98-NL-127,(1998) 、参照）。ここでは「といったこと」の例で説明する。
【００３６】
「といったこと」を解析し、juman のＡというバージョンの出力が

と	と	助詞
いった	言う	動詞
こと	こと	名詞

となっていて、Ｂのバージョンの出力が

と	と	助詞
いった	行く	動詞
こと	こと	名詞

となっているとする。「いった」という語は「行く」と「言う」の曖昧性があり、Ｂのバ

ージョンではこれを誤って「行く」の方の語であると出力していたとする。ここでmdiffをとると以下のような結果となる。

【００３７】

と　と　助詞
；▽▽▽▽▽▽
いった　言う　動詞
；●●●
いった　行く　動詞
；△△△△△△
こと　こと　名詞

mdiff をとることで複数のシステムの出力の差異を容易に検出することができる。この場合、「いった」の部分が出力に差異があることが分かる。ここで、出力修正の作業者は、このような差分が検出された箇所においてどちらが正しいかを判断し、上が正しければ何もせず下が正しければ「；●●●」の先頭に“ｘ”を付けるなどとすると決めておく。そのようにすると、“ｘ”がなければ差分の下を、あれば差分の上の情報と区切り、記号を消すことで、その作業結果のデータから自動的にそれぞれの差分からよい結果の方を選び、それぞれのバージョンのものより高い精度の結果を生成できる。また、差分の両方が誤っている場合がよくある。このときは「；●●●」の上の方のデータを実際に書き直すとよい。

【００３８】

この方法を用いると、修正できないものは両方のバージョンで同じように誤るものだけであり、多くの形態素誤りを修正できる。ここで注意すべきことは異なる性質のシステムを複数用意しないといけないということである。誤り方が同じシステムの場合だと多くの誤りを見逃すことになる。

【００３９】

また、システムが三つある場合は、diff3 コマンドを使うとよい。diff3 は、三つのファイルの差分を検出することができる。

【００４０】

上記では、形態素解析を例にあげたが、他の解析でも解析結果を行単位にすることでmdiff で差分をとることができる。また、文字単位が必要ならばmdiffcを使えばよい。

【００４１】

ここでは、複数のシステムの出力の差分をとる話をしたが、一つをタグ付きコーパスとし、それを何かのシステムで解析した結果と比較することで、そのタグ付きコーパスの誤りを検出し修正することもできる。

【００４２】

(2) 差分の考察と書き換え規則の獲得の説明

ここでは、話し言葉と書き言葉のdiffの研究について記述する。対応のとれた話し言葉と書き言葉のデータを使い、それらの差分から話し言葉と書き言葉の違いを考察したり、話し言葉から書き言葉への言い換え規則、また、その逆のための規則を獲得するものである。データとしては、学会の口頭発表を話し言葉データとし、その口頭発表の内容が記されたその学会の予稿原稿を書き言葉として用いた。

【００４３】

図４は書き言葉データと話し言葉データの例の説明図である。例えば、書き言葉と話し言葉のデータが図４のような形で与えられたとする。ここでは、差分がとりやすいように形態素解析システム（形態素解析手段３）などで１行に１単語がはいるような形に変換してある。このような書き言葉と話し言葉のデータが与えられたとき、mdiff をとると、図５のような結果を得る。図５は書き言葉データと話し言葉データのdiffの結果の説明図である。図５の結果から、差分部分だけを抽出すると、図６のような結果が得られる。図６は差分部分の抽出の説明図である。

【００４４】

図６の結果から、話し言葉には「え」などが挿入されること、また話し言葉では「っていうの」という表現をいれて発話をなめらかにすることなどが分かる。また、「述べる」が「述べます」と言い換えられていることが分かる。以上のように、mdiff を使うことで話し言葉と書き言葉の差異を検出でき、また、それを考察することで、話し言葉と書き言葉の違いのようなものを調査できることが分かる。また、これらの差分は、話し言葉と書き言葉の言い換え規則としてみることもできる。

【００４５】

例えば、「え」の部分は、書き言葉に何もないところに話し言葉に変換する場合「え」をいれるという規則のように見ることができる。また、「述べる」と「述べます」の部分は、話し言葉に変換する場合は「述べる」を「述べます」に言い換える規則のように見ることができる。その意味でmdiff を用いることで言い換え規則、もしくは、変換規則のようなものを検出できることが分かる。

【００４６】

ここでは、話し言葉と書き言葉のデータを例にとったが、このようなことは様々なところで可能である。例えば、英文校閲前のテキストと英文校閲後のテキストで、mdiff をとると、どのような違いをどのように直せばよいかが分かるし、また英文校閲用の規則のようなものが獲得できる。また、要約前のテキストと要約後のテキストで、mdiff をとると、どのように要約されているかを如実に見ることができるし、要約用の規則のようなものが獲得できる。その他にも対応のとれた性質の異なるデータに対してmdiff をとると、様々な考察と、言い換え規則の獲得ができる。

【００４７】

（４）：データのマージの説明

(1) 対訳コーパスの対応付けの説明

ここでは、対訳コーパスの対応付けを考える。条件として、それぞれのコーパスには、対応する箇所に同じ記号が入っていることを前提とする。また、対応付けの単位は、この記号で区切られた部分であるとする。

【００４８】

図７はコーパスの構成の説明図である。ここでは、日本語のコーパスと英語のコーパスがまだ、ばらばらに存在し、対応付けられていないとする。図７の例のように両方ともSection 1 などの同じ形をしたセクション情報が与えられているとする。このとき、日本語と英語では、同じセクションのものは、同じ内容であるとする。

【００４９】

この場合、これらのデータのmdiff をとることで、図８のような結果を得ることができる。図８はmdiff によって対応付けられた対訳コーパスの説明図である。図８の結果では、Section 1 などが共通部分となり、その他の部分が不一致部分となる。この不一致部分では、日本語と英語が上下に分かれて格納されることになる。このようにすることで、mdiff を用いて対訳データが作成されることになる。

【００５０】

ここで示したものは、文ごとなどの細かい対応付けをするものではなく、セクションなどの大雑把なもので一見役に立たないように思われるかもしれないが、文の対応付けは難しい問題で、まず予め対応がとれていることがはっきりしている章、段落のレベルで対応付けをしてから細かい対応付けをするという考え方もあり、その意味ではこのような粗い対応付けも役立つものである。

【００５１】

また、ここで示したものは、Section 1 などの情報を認識させて区分するだけでそのようなことをするプログラムを書くことでも同じように対訳データの対応付けを行うことができる。しかし、mdiff を使うとそのような複雑なプログラムを書くこともなく対応付けを容易に実現できるものである。

【００５２】

図９は対訳コーパスの言語処理システムの説明図である。図９において、対訳コーパス

の言語処理システムには、mdiff 処理部２１が設けてある。mdiff 処理部２１は、入力された原文データと翻訳データとの二つのファイルのmdiff を出力するものである。

【００５３】

図１０は対訳コーパスのmdiff によるフローチャートである。以下、図１０の処理Ｓ１１～Ｓ１３に従って説明する。

【００５４】

Ｓ１１：mdiff 処理部２１に、入力として二つのファイルが与えられる。ここではこの二つのファイルは、それぞれ英語文章、日本語文章を格納したものである。

【００５５】

Ｓ１２：mdiff 処理部２１で、この二つのファイルのmdiff をとる。

【００５６】

Ｓ１３：mdiff 処理部２１は、mdiff の結果を出力する。

【００５７】

(2) 講演と予稿の対応付けの説明

講演と予稿の対応付けを考える。この講演と予稿は、先の書き換え規則の獲得でも述べた書き言葉データと話し言葉データに対応する。即ち、講演は学会の口頭発表で、予稿はその口頭発表に対応する論文のことである。このような講演と予稿が与えられたとき、講演の各部分と、予稿の各部分の対応がとれると、講演を聞いている時だと、それに対応する予稿の部分を参照できるし、予稿を読んでいるときだと、それに対応する講演の部分を参照できて便利である。ここでは、この講演と予稿の対応付けをmdiff で行う説明をする。

【００５８】

ここでは、特に予稿の各章が講演のどこの部分に対応するかをmdiff でもとめることにする。ここで予稿と講演とは、同じ順序でなされると仮定する。また、予稿の章が認識しやすいように予稿データには、図１１のように“<Chapter 1> ”のような記号を挿入しておく。図１１は予稿データの構成の説明図である。この形にしておいて、予稿と講演のデータに対して、形態素解析をして各行に単語がくる状態でmdiff を使うことで、もしくは、mdiffcを使うことで、図１２のような結果を得る。図１２は予稿と講演のmdiff の結果の説明図である。ここで、差分部分で予稿に対応する上半分の方を、“<Chapter 1> ”のような記号を除いてすべて消し去ると図１３のような結果を得る。図１３は講演データへの章の情報の挿入結果の説明図である。図１３では、元の講演データに対して“<Chapter 1> ”のような記号だけが挿入された形になる。つまり、講演のどの部分が予稿のどの章にあたるかが分かることになる。

【００５９】

これは簡単にいうと、mdiff の照合能力を用いて予稿と講演を照合し、章の情報だけ残して予稿の情報を消し去ることにより、講演データに章の情報を挿入するということを行っていることを意味する。このような予稿と講演の対応付けもmdiff を用いると簡単に行うことができる。

【００６０】

図１４は講演と予稿の対応付けの言語処理システムの説明図である。図１４において、mdiff の言語処理システムには、mdiff 処理部２１、予稿削除部２２が設けてある。mdiff 処理部２１は、入力された二つのファイルのmdiff をとり出力するものである。予稿削除部２２は、予稿側の差分部分で<Chapter 1> などの章情報のみを残して予稿データをすべて削除するものである（mdiff 記号の“；△△△△△△”なども削除する）。

【００６１】

図１５は講演と予稿の対応付けの処理フローチャートである。以下、図１５の処理Ｓ２１～Ｓ２４に従って説明する。

【００６２】

Ｓ２１：mdiff 処理部２１に、入力として二つのファイルが与えられる。この二つのファイルは、それぞれ予稿、講演の文章を格納したものである。また、予稿データの方は、

<Chapter 1> などの章の範囲を示す記号が付されているものとする。

【００６３】

Ｓ２２：mdiff 処理部２１で、この二つのファイルのmdiff をとる。

【００６４】

Ｓ２３：予稿削除部２２は、予稿側の差分部分で<Chapter 1> などの章情報のみを残して予稿データをすべて削除するものである。また、mdiff 記号の“；△△△△△△”なども削除する。

【００６５】

Ｓ２４：予稿削除部２２は、予稿側の差分部分で章情報のみを残して予稿データをすべて削除した結果を出力する。

【００６６】

（５）：最適照合能力を用いた質問応答システムの説明

ここでは mdiffの最適照合能力を用いた質問応答システム（質問応答言語処理システム）について記述する。質問応答システムとは、例えば、「日本の首都はどこですか」と聞くと「東京」と答えそのものをずばり返すシステムである。

【００６７】

知識が自然言語で書かれていると仮定すると、基本的には質問文と知識の文を照合し、その照合結果で疑問詞に対応するところを答えとして出力すればよい。例えば先の問題だと、「日本の首都は東京です」という文を探してきて、この文で疑問詞に対応する「東京」を解として出力するのである。ここではこれを mdiffで行なうことを考える。

【００６８】

まず、質問文の疑問詞の部分をＸに置き換え、また文末を平叙文に変換し、「日本の首都はＸです」を得る。また、知識ベースから「日本の首都は東京です」を得る。ここでこの二つの mdiffc をとると以下のような結果を得る。

【００６９】

日
本
の
首
都
は
；▽▽▽▽▽▽
X
；●●●
東
京
；△△△△△△
で
す

ここでＸと差分部分で組になっているものを解とすると、「東京」を正しく取り出せることになる。

【００７０】

ところで mdiffc を使う場合少々文に食い違いがあっても、答えを正しく取り出すことができる。例えば、知識ベースの文が「日本国の首都は東京です」であったとする。この場合は mdiffc の結果は以下のようになる。

【００７１】

日
本
；▽▽▽▽▽▽
；●●●

国
；△△△△△△
の
首
都
は
；▽▽▽▽▽▽
X
；●●●
東
京
；△△△△△△
で
す

差分部分は少し増えるがＸに対応する箇所は「東京」のままで、解を正しく抽出できる。

【００７２】

ところで、われわれが提案する質問応答システムでは、類似度を尺度として用いた変形をくりかえし、質問文と知識データの文がより一致した状態で上記のような照合を行なう。このために類似度を定義する必要がある。

【００７３】

mdiff を用いた場合は、一致部分と不一致部分が認定できるので、類似度は、（一致部分の文字数）／（全文字数）のような形で定義できる。ここではmdiff により類似度を求めるようなことをしている。このように mdiffは文の類似性／類似度を求めることにも役に立つ。

【００７４】

ここで、「日本国」と「日本」を言い換える規則があれば「日本の首都はＸです」を「日本国の首都はＸです」と言い換えて照合し、不一致部分を減らすことで、より確実に解を得ることができる。

【００７５】

このように、 mdiffを使うだけで簡単な質問に応答する言語処理システムを、容易に構築できることは簡便さの観点から価値がある。

【００７６】

図１６は質問応答システムの説明図である。図１６において、質問応答システムには、質問文変換部３１、質問文保存部３２、キーワード抽出部３３、データベース文検索部３４、データベース文保存部３５、類似度演算部３６、mdiff 処理部３７、質問文変形部３８、データベース文変形部３９、対応部出力部４０が設けてある。質問文変換部３１は、疑問文を平叙文にまた疑問詞をＸに変換するものである。質問文保存部３２は、質問文を保存するものである。キーワード抽出部３３は、質問文からキーワードを抽出するものである。データベース文検索部３４は、キーワードを多く含むデータベース文を検索するものである。データベース文保存部３５は、データベース文検索部３４で検索したデータベース文を保存するものでる。類似度演算部３６は、質問文とデータベース文とを比較し類似度を求めるものである。mdiff 処理部３７は、質問文とデータベース文とを１文字単位でmdiff をとるものである。質問文変形部３８は、質問文の変形を行うものである。データベース文変形部３９は、データベース文の変形を行うものである。対応部出力部４０は、mdiff 処理部３７の出力結果のうち“Ｘ”に対応するデータベース側の表現を抽出し出力するものである。

【００７７】

図１７は質問応答の処理フローチャートである。以下、図１７の処理Ｓ３１～Ｓ３９に従って説明する。

【００７８】
Ｓ３１：質問応答システムの類似度Ｓ０の値を適当に与え（通常最初は“０”とする）処理Ｓ３２に移る。
【００７９】
Ｓ３２：質問応答システムの入力として、質問文が与えられ処理Ｓ３３に移る。
【００８０】
Ｓ３３：質問文変換部３１は、与えられた質問文の疑問文を平叙文に、また疑問詞をＸに変換し、この変換結果の文を質問文保存部３２に渡す。
【００８１】
Ｓ３４：キーワード抽出部３３にも、入力として与えられた質問文が渡され、質問文からキーワードが抽出される。この抽出されたキーワードはデータベース文検索部３４に渡され処理Ｓ３５に移る。
【００８２】
Ｓ３５：データベース文検索部３４で、キーワードを多く含むデータベース文が、知識ベース（図示しないデータベース）からいくつか検索され、その結果はデータベース文保存部３５に格納され処理Ｓ３６に移る。
【００８３】
Ｓ３６：類似度演算部３６で、質問文とデータベース文とを比較し類似度を求める。この類似度の算出にはmdiff 処理部３７を用いる。即ち、類似度を求めたい質問文とデータベース文をmdiff 処理部３７に入力し、このmdiff 結果により一致部分の文字数、不一致部分の文字数が求まる。類似度は、（一致部分の文字数）／（全文字数）と予めきめておく。そして、この類似度を質問文保存部３２とデータベース文保存部３５にあるすべての文の組に対して求め、このとき最も類似度が高かったときの類似度をＳとする。次に、類似度演算部３６は、類似度ＳがＳ０より大きいかどうか判断する。この判断で、類似度ＳがＳ０より大きい場合は処理Ｓ３７に移り、類似度ＳがＳ０と同じとき処理Ｓ３８に移る。
【００８４】
Ｓ３７：質問文変形部３８とデータベース文変形部３９では、それぞれ質問文とデータベース文の変形を行い、その変形結果をそれぞれの保存部に格納する。類似度演算部３６はＳ０に類似度Ｓの値をセットし、処理Ｓ３６に戻る。なお、ここでの変形とは、「日本国の首都はＸです」を「日本の首都はＸです」にいいかえるようなことを意味している。
【００８５】
Ｓ３８：類似度演算部３６は、このときの類似度Ｓの値を求めるときに使ったmdiff 処理部３７の出力結果を対応部出力部４０に渡し処理Ｓ３９に移る。
【００８６】
Ｓ３９：対応部出力部４０では、mdiff 処理部３７の出力結果のうち“Ｘ”に対応するデータベース側の表現を抽出して出力する。
【００８７】
なお、上記実施の形態では、差分検出を行うのにdiffコマンドを用いたが、予め定めた単位で、順序情報を保持したまま一致部分を最大にする対応付けを行うシステムであれば他の差分検出手段を用いることができる。
【００８８】
（６）：プログラムインストールの説明
入力手段１、処理部２、形態素解析手段３、差分検出手段４、類似度演算手段５、ＵＮＩＸdiff処理部１１、整形部１２、mdiff 処理部２１、予稿削除部２２、質問文変換部３１、質問文保存部３２、キーワード抽出部３３、データベース文検索部３４、データベース文保存部３５、類似度演算部３６、mdiff 処理部３７、質問文変形部３８、データベース文変形部３９、対応部出力部４０等は、プログラムで構成でき、主制御部（ＣＰＵ）が実行するものであり、主記憶に格納されているものである。このプログラムは、一般的な、コンピュータで処理されるものである。このコンピュータは、主制御部、主記憶、ファ

イル装置、表示装置、キーボード等の入力手段である入力装置などのハードウェアで構成されている。

【００８９】

このコンピュータに、本発明のプログラムをインストールする。このインストールは、フロッピィ、光磁気ディスク等の可搬型の記録（記憶）媒体に、これらのプログラムを記憶させておき、コンピュータが備えている記録媒体に対して、アクセスするためのドライブ装置を介して、或いは、ＬＡＮ等のネットワークを介して、コンピュータに設けられたファイル装置にインストールされる。そして、このファイル装置から処理に必要なプログラムステップを主記憶に読み出し、主制御部が実行するものである。

【図面の簡単な説明】

【００９０】

【図１】本発明の言語処理システムの説明図である。

【図２】実施の形態におけるmdiff の言語処理システムの説明図である。

【図３】実施の形態におけるmdiff によるフローチャートである。

【図４】実施の形態における書き言葉データと話し言葉データの例の説明図である。

【図５】実施の形態における書き言葉データと話し言葉データのdiffの結果の説明図である。

【図６】実施の形態における差分部分の抽出の説明図である。

【図７】実施の形態におけるコーパスの構成の説明図である。

【図８】実施の形態におけるmdiff によって対応付けられた対訳コーパスの説明図である。

【図９】実施の形態における対訳コーパスの言語処理システムの説明図である。

【図１０】実施の形態における対訳コーパスのmdiff によるフローチャートである。

【図１１】実施の形態における予稿データの構成の説明図である。

【図１２】実施の形態における予稿と講演のmdiff の結果の説明図である。

【図１３】実施の形態における講演データへの章の情報の挿入結果の説明図である。

【図１４】実施の形態における講演と予稿の対応付けの言語処理システムの説明図である。

【図１５】実施の形態における講演と予稿の対応付けの処理フローチャートである。

【図１６】実施の形態における質問応答システムの説明図である。

【図１７】実施の形態における質問応答の処理フローチャートである。

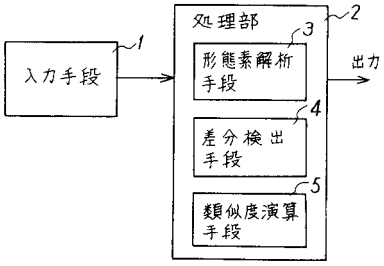
【符号の説明】

【００９１】

１　入力手段
２　処理部
３　形態素解析手段
４　差分検出手段
５　類似度演算手段

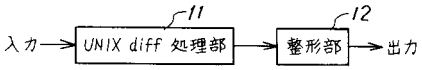
【図１】

言語処理システムの説明図



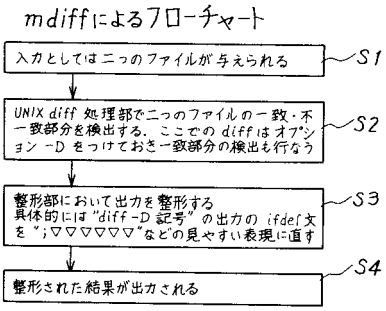
【図２】

mdiffの言語処理システムの説明図



【図３】

mdiffによるフローチャート



【図４】

書き言葉データと話し言葉データの例の説明図

書き言葉データ	話し言葉データ
本	今日
論文	は
で	え
は	意味
意味	ソート
ソート	に
に	ついて
ついて	述べ
述べる	ます
。	一般に
一般に	ソート
ソート	って
は	いう
50	の
音	は
順	だいたい

【図５】

書き言葉データと話し言葉データのdiffの結果の説明図

;▽▽▽▽▽▽	ついて
本	;▽▽▽▽▽▽
論文	述べる
で	。
;●●●	;●●●
今日	述べ
;△△△△△△	ます
は	;△△△△△△
;▽▽▽▽▽▽	一般に
;●●●	ソート
え	;▽▽▽▽▽▽
;△△△△△△	;●●●
意味	って
ソート	いう
に	の
(右欄につづく)	;△△△△△△

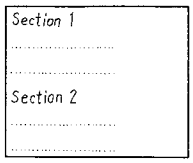
【図６】

差分部分の抽出の説明図

書き言葉データ	話し言葉データ
本論文で	今日
	え
述べる。	述べます
	っていうの

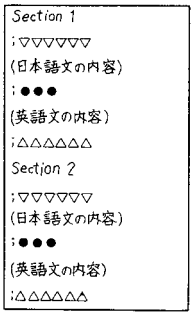
【図７】

コーパスの構成の説明図



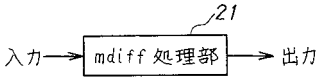
【図８】

mdiffによって対応付けられた対訳コーパスの説明図



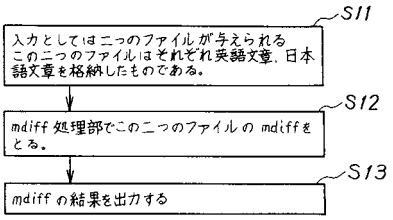
【図９】

対訳コーパスの言語処理システムの説明図



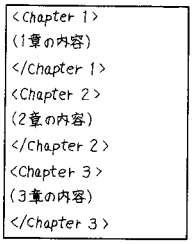
【図１０】

対訳コーパスのmdiffによるフローチャート



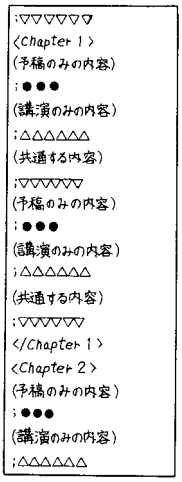
【図１１】

予稿データの構成の説明図



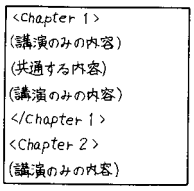
【図１２】

予稿と講演のmdiffの結果の説明図



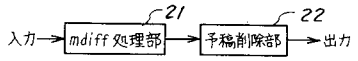
【図１３】

講演データへの章の情報の挿入結果の説明図



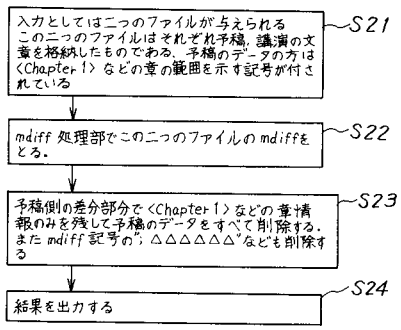
【図１４】

講演と予稿の対応付けの言語処理システムの説明図



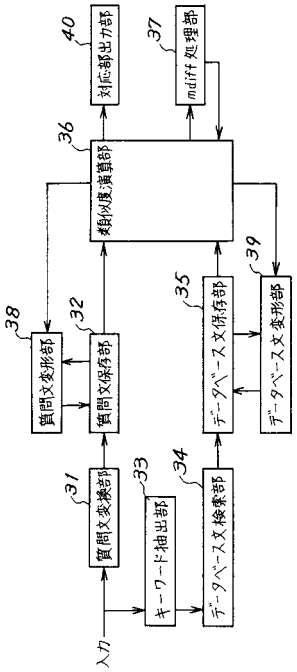
【図１５】

講演と予稿の対応付けの処理フローチャート



【図１６】

質問応答システムの説明図



【図１７】

質問応答の処理フローチャート