自 然 言 語 処 理 106－5
（1995．3．9）

オンラインマニュアル概要文を利用した異機種間の類似コマンド検索方法

A Method for Retrieving Similar Commands from Different Computers Using Command Summaries

安達 久博　下山 豪彦

Hisahiro Adachi and Hidehiko Shimoyama

宇都宮大学工学部情報工学科

Department of Information Science, Utsunomiya University

あらまし

近年，計算機ネットワークに多機種の計算機を接続した使用環境が整備されつつある．しかし，機種によりOSのバージョンが異なるために，例えば，UNIXを基本OSとしたワークステーションでは，同一機能を提供するコマンド名が機種により異なる場合や同じコマンド名であるにもかかわらず，提供する機能が異なる等の問題点が指摘されている．これらの問題にユーザが直面した場合，計算機システムが提供しているコマンドの検索機能は，同一機種のコマンドだけを対象としており，異機種間に対応していないため，ユーザが対応するコマンドを調べる負荷は非常に大きい．本稿では，上記の問題点を解消するため，各コマンドの機能を要約記述したマニュアル概要文間の類似性に着目し，コマンド間の類似性をこの概要文間の類似性で近似し，異機種間の類似コマンドを検索する方法を提案する．

The UNIX operating system was one of the first to include online documentation. The online manual is complete, authoritative, and usually more current than any printed documentation. Unfortunately, There are two slightly different versions: one for Berkeley-based systems, one for System V. Therefore, the same command name often provides the different functions. In this paper, we propose a method for retrieving similar commands from different computers using command summaries. The method is based on the similarity between command manual summaries. The measure of similarity between command manual summaries is derived from their common words as a set-theoretical measure. Therefore, this method is not necessary the morphological analysis. The results of experiments are that we can be retrieved the significant similar commands from different computers.

1. はじめに

近年，計算機ネットワークに異機種の計算機を接続した利用環境が整備されつつある．この様な計算機ネットワーク環境下では，異なる機種間で使用するコマンド名やその機能が異なるなどの問題点が指摘されている．

例として，UNIXを基本OSとする計算機システムを考えてみよう．A社で動作しているプログラムをB社の計算機に移植する場合，プログラムで使用しているライブラリ関数の名前がB社の計算機では別の名前で提供されていたり，関数名は同じでもその機能が異なるなどの問題である．

これらの問題は，ユーザが一般に使用するコマンドでも同様である．例えば，一般に「リモートシェルコマンド」は，「rsh」という名前のコマンドであるが，別の計算機では「rsh」は「限定(restricted)シェル」を意味し，コマンドの機能が全く異なる．一方，「remsh」と命名されたコマンドが「リモートシェル」の機能を提供している．

ところで，ユーザがコマンドやライブラリ関数を調べる場合，膨大な量の冊子体の計算機マニュアルが提供されているが，検索，参照の煩わしさから最近は，オンライン電子化マニュアルを利用する傾向にある．一般に，このオンラインマニュアルを検索表示するためのコマンドとして，ほとんどのUNIX計算機は，「man」コマンドを提供している．

図1.は，manコマンドを使用してマニュアルを検索，表示した例である．

また，この「man」コマンドには，オプション機能として以下に示す2つのコマンド検索機能を

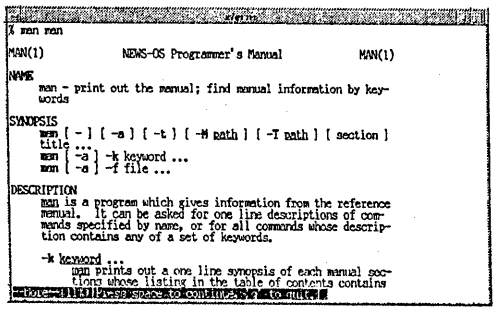

図 1: man コマンドの表示例

提供している．これらの機能は，オンラインマニュアルから，コマンド名とその機能を 1 行文に要約した「マニュアル概要文」を抽出した whatis データベースを使用している．図 1.の「NAME」欄に示された文を要素とする有限の文集合と捉えることができる．図 2. に，whatis データベースの一部を示す．

```
touch (1)                - update date last modified of a file
tp (1)                   - manipulate tape archive
tr (1)                   - translate characters
traceroute (1)           - print the route packets take to network
traper (3F)              - trap arithmetic errors
trek (6)                 - trekkie game
troff, nroff (1)         - text formatting and typesetting
trpfpe, fpecnt (3F)      - trap and repair floating point faults
trpt (8C)                - transliterate protocol trace
true, false (1)          - provide truth values
truncate, ftruncate (2)  - truncate a file to a specified length
tset (1)                 - terminal dependent initialization
```

図 2: whatis データベースの内容例

man の検索機能

- **man -f**

  指定されたコマンドあるいはライブラリ関数名に対応するマニュアルの存在場所（セクション番号）と，その機能を簡潔に表現したマニュアル概要文の検索表示機能．

- **man -k**

  指定されたキーワード（英単語）をコマンドあるいはライブラリ関数名とそのマニュアル概要文のどちらかに含む全てのコマンド名の検索表示機能．キーワードは複数指定できるが膨大な量の検索結果を表示する場合がある．

図 3は，これらの機能の使用例である．

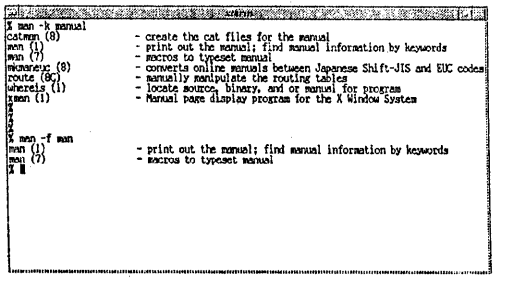

図 3: man のオプション機能の実行例

これらの検索機能は，ユーザが現在使用している計算機のコマンドに対してのみ有効であり，上記のような異機種間でのコマンド検索機能は提供されていないのが現状であり，異機種での対応コマンドを調べる際のユーザの負荷は大きく，問題点として指摘されている．

本稿では，異機種間のコマンドの類似性を対応するマニュアル概要文間の類似性と捉え，概要文間の類似度に基づくコマンドの類似検索方法を提案する．

2 章では，本研究の対象データである UNIX のマニュアル概要文の記述形式の特徴と異機種間での相違点について述べ，3 章では，マニュアル概要文間の類似度を定義し，4 章では，異機種間でのコマンドの類似検索方法を述べ，5 章で実験と検討を行い，6 章でまとめを行う．

2. コマンド間の類似性

前節で議論したように，異機種間のコマンドの類似性を調べる場合，コマンド名の類似性だけからは判断ができない．(例えば，rsh と remsh の関係）そこで，本稿では，コマンド間の類似性の尺度を対応するマニュアル概要文間の類似性の尺度で近似する方針とする．以下，マニュアル概要文の特徴とその類似度の計算方法について述べる．

2.1 マニュアル概要文の特徴

マニュアル概要文（以後，概要文と略記する）は，コマンドの機能を 1 行文（平均単語数約 10 個）で簡潔に表現した英語文である．また，一般の英語文に比べ，計算機のコマンドの機能を説明する性質上，文中で使用される単語集合や文法パ

ターンに制約がある文集合と捉えることができる．特に，本稿が対象とする異機種の計算機は，バージョンの違いはあるものの基本 OS として UNIX を採用しているため，異機種間のコマンドに対応する概要文間の類似性は高く，同一の概要文も存在する．

本稿では，UNIX を基本 OS とする異機種の計算機システムとして，System V 系 OS の計算機（以後 A 計算機と略記する）と BSD 系 OS の計算機（以後 B 計算機と略記する）間の概要文の比較により記述形式の特徴を述べる．尚，A 計算機は HP-UX，B 計算機は SONY-NEWS とし，それぞれの whatis データベース中のセクション番号 1（一般ユーザが使用できるコマンド）に分類されるコマンドを対象とする．

ここでは，文献 [1] に示されている両方の計算機で共通のコマンド（コマンド名とその機能が共通）73 個について，A 計算機と B 計算機の whatis データベースから抽出したマニュアル概要文を使用し，比較を行った．その結果，同一の概要文を持つコマンドは 36 個（約 49 %），また，その他の概要文間では，以下のような相違点があった．

- **大文字と小文字**

  例．コマンド f77

  A 計算機: FORTRAN 77 compiler

  B 計算機: Fortran 77 compiler

- **ハイフンの有無**

  例．コマンド　tset

  A 計算機: terminal-dependent initialization

  B 計算機: terminal dependent initialization

- **単（複）数形と冠詞**

  例．コマンド　pr

  A 計算機: print files

  B 計算機: print file

- **単語の挿入，追加**

  例．コマンド　tar

  A 計算機: tape file archiver

  B 計算機: tape archiver

- **単語の語順**

  例．コマンド　roff,nroff

  A 計算機: format text

  B 計算機: text formatting

- **複合要因**

  例．コマンド　stty

  A 計算機: set the options for a terminal port

  B 計算機: set terminal options

これらの相違点は単独で出現するのではなく，組み合せ的に「複合要因」となるものがほとんどである．以上の相違点を考慮し，次節でマニュアル概要文間の類似度を定義する．

3. マニュアル概要文間の類似度

従来，文間の類似性を判断する方法として，形態素解析等の本格的な自然言語処理を駆使した方法が提案されている．一方，文を文字列と捉えたパターンマッチング処理を利用した 2 つの文字列間の類似性を計算する尺度として，文字の挿入，脱落，置換操作のコストの最小値に基づく Tai の距離や文字の順序を保存する形で最長共通部分文字列（LCS）に基づく距離あるいは類似度を DP 法で計算する方法がある．

本稿では，本格的な自然言語処理技術を使用しない方針で議論を進める．また，前節で議論した様に対象文の特徴から単語の挿入脱落や語順の逆転現象があるため，上記の計算方法も採用できないと考える．

そのため，本稿では情報検索の分野で広く利用されているキーワードの集合論的尺度を概要文間の類似度として採用する．形式的には，概要文間の類似度は，以下の式で計算される．

概要文間の類似度

2 つのマニュアル概要文を $A = a_1 a_2, ..., a_m$, $B = b_1 b_2, ..., b_n$ とする．ここで，$a_i (1 \leq i \leq m), b_j (1 \leq j \leq n)$ は，文の構成要素である単語を意味する．また，文 A の単語集合を $X_A = \{a_1, a_2, ..., a_m\}$ で表わし，文 B の単語集合を $X_B = \{b_1, b_2, ..., b_n\}$ で表わすとマニュアル概要文間の類似度 $S(A, B)$ は，次式で計算される．

$$S(A,B) = \frac{num(X_A \cap X_B)}{num(X_A \cup X_B)} \qquad (1)$$

$$(0 \leq S(A,B) \leq 1)$$

ここで，$num(X_A \cap X_B)$ は共通集合の要素数であり，$num(X_A \cup X_B)$ は和集合の要素数である．

また，この類似度は反射律 ($S(A,A) = 1$)，対称律 ($S(A,B) = S(B,A)$) を満たすことは明らかである．

次に，集合要素を計算する場合に前節で議論した概要文間の相違点を考慮して，以下の単語照合基準に従って，集合要素を計算する．

単語の照合基準

1. 冠詞，前置詞，接続詞を不要語 (stop word) とする．

   例えば，「change mode file」と「change mode of a file」の同一視．

2. 単語の先頭から連続して4文字以上が一致し，かつキーとなる単語の文字数と一致した文字数の割合が80%以上の場合，同一単語とみなす．

   例えば，「line」と「lines」や「directory」と「directories」などの単数形と複数形の同一視，「link」と「linc」の非同一視．また，「generate program for ...」と「generator of ... programs」の様な派生語の同一視．

3. 大文字と小文字の区別をしない

   例えば，「Fortran」と「FORTRAN」，「UNIX」と「unix」等の同一視．

4. ハイフンやスラッシュは空白に置換し単語分割を行う．

   例えば，「C program checker/verifier」と「C program verifier」の同一視．「screen-oriented display」と「screen oriented display」の同一視．

4. コマンドの類似検索方法

ここでは，A計算機のコマンド名をキーにしてB計算機の類似コマンドを検索する場合を例に説明する．異機種間の類似コマンドの検索手順は以下の通りである．

類似検索手順

- (1) 計算機Aのコマンド名を入力する．
- (2) 入力されたコマンド名に対応する計算機Aの概要文を検索する．
- (3) (1)で指定されたコマンド名と同一のコマンド名を持つ計算機Bの概要文を検索する．
- (4) (2)と(3)で得られた概要文間の類似度を計算し，類似度=1ならば検索を終了し，それ以外は(5)へ
- (5) (2)で検索された概要文と計算機Bの全ての概要文間の類似度を計算し，類似度が高い順に概要文データをソートする．
- (6) (1)で指定されたコマンド名と最高類似度の概要文に対応するコマンド名を比較し，同一なら検索終了し，それ以外は(7)へ．
- (7) 類似度の上位から順に概要文データに対応するコマンドを表示する．

5. 実験と検討

5.1 準備

実験の対象コマンドとして，文献[2]に記載のBSD系OSとSystem V系OSに共通のコマンドリストから100個を選択し，計算機Aと計算機Bの対応する概要文をwhatisデータベースから抽出し，2つの試験データAとBを準備した．

また，A計算機とB計算機のwhatisデータベースからセクション番号(1)に分類されているコマンドとその概要文をペアにしたリストを作成した．尚，window関係やグラフィック関係のコマンドは対象外とした．その結果，計算機Aの概要文データベースとして309エントリ，計算機Bの概要文データベースとして395エントリを実験対象データベースとした．

5.2 実験方法と結果

実験.1

実験方法は，まず最初に上記の準備で用意した100個のコマンドに対して，試験データAと計算機Bの概要文間の類似度を計算し，上位から何番目に同一のコマンド名を持つ概要文が検索されたかで評価を行う．同様に試験データBと計算機Aの概要文間についても実験を行った．そのの結果を表1に示す．

	1位で検索	2位で検索	3位以下で検索
AからBの検索	42+44 = 86	8	6
BからAの検索	42+42 = 84	6	10
平均検索率	85 %	7 %	8 %

表 1: 実験.1の結果

ここで，表中の第1位で検索されたコマンドで42個は同一概要文が存在した（正確には，先に述べた単語照合基準により同一視され，類似度が1の場合も含む）．この結果，双方向からの検索で上位2位以内で該当コマンドが検索された割合は平均92％となる．第3位以下で検索されたもののほとんどは類似度が0，つまり概要文間に共通単語が存在しない場合である．詳細な内容は検討の項で述べる．

実験.2

次に，計算機Aと計算機Bの概要文データベースを対象として，全組み合せの類似度を計算し，上位のペアの中でコマンド名が異なり，その機能が同一あるいは類似のコマンドの例を表2に示す．

計算機A	batch	xdb	xd	remsh	adjust	red	cb	pg
計算機B	at	dbx	hd	rsh	fmt	ed	indent	page

表 2: 類似コマンドの例

5.3 検討

ここでは，実験.1において上位3位以下で該当コマンドを検索した例を示しながら，その原因について検討を行う．

本稿で提案した類似検索方法は，概要文間の共通単語の個数をキーにしているため，以下の例の様に共通単語が存在しない場合やほとんど無い場合に類似度が低下し，上位で検索不能となる．

例.1

```
su = become super-user or another user
1 (0.40) write = write to another user
2 (0.40) talk = talk to another user
3 (0.17) yacc = yet another compiler-compiler
4 (0.17) wall = write to all users
5 (0.14) finger = user information lookup program
○6 (0.14) su = substitute user id temporarily
```

```
cat = concatenate, copy, and print files
1 (0.50) pr = print file
2 (0.40) fpr = print Fortran file
3 (0.40) dd = convert and copy a file
4 (0.40) rcp = remote file copy
5 (0.33) showsnf = print contents of an SNF file
○11 (0.20) cat = catenate and print
```

また，次の例は計算機Bで単一コマンドに概要文が記述されているのに対して，計算機Aでは，複数の関連するコマンドを結合し，概要文が記述されているために上位で検索されない事例である．

例.2

```
cp,ln,mv = copy, link or move files
1 (0.40) rcp = remote file copy
2 (0.40) dd = convert and copy a file
3 (0.40) mv = move or rename files
4 (0.33) cpio = copy file archives in a
5 (0.25) cp = copy
(END)
```

これら概要文間の差は，概要文の類似性に基づく本方式の限界を示している．

これらの問題点を解消するためには，単語の照合方法に単語間のシソーラスや文解析技術などの本格的な自然言語処理技術の利用を検討する必要がある．一方，検索順位を重視するのでは無く，

同一名のコマンドが存在するか否か，また存在する場合に，異機種間で概要文の比較ができる情報を提供できる機能は従来は提供されておらず，本方式は有効に対処できると考える．

6. おわりに

本稿では，異なる計算機間での類似コマンドの検索方法を提案した．まず，コマンドの機能の類似性をコマンドに対応するマニュアル概要文間の類似性で近似し，概要文間の類似度は構成単語の共通集合の割合に基づく集合論的尺度を採用した．実験の結果，類似の概要文を持つコマンドの平均検索率は上位 2 位以内で 92 %の結果を得た．しかし，機種により概要文の記述の差が大きい場合（単語や文体）に不適切な検索結果を提示する本方式の限界がある．

今後の課題としては，既存の whatis データベースを利用せずにオンラインマニュアルから独自の検索用データベースの構築と利用方法の検討がある．また，日本語のマニュアル概要文を利用することで機種による概要文間の差を検討し，英語文を利用した本方式との比較を行う予定である．

参考文献

[1] 井田昌之:UNIX 詳説 --- 基礎編 ---，丸善,1984.

[2] 坂本文:UNIX ツールガイドブック，共立出版,1986.

[3] 長尾眞:パターン情報処理，コロナ社,1983.

[4] 飯島泰蔵:パターン認識理論，森北出版,1989.

[5] 伊藤哲郎:情報検索，昭晃堂,1986.

[6] 安達久博，下山豪彦:"多機種間のマニュアル概要検索のためのファジー検索拡張シェル freshman"，情報処理学会第 49 回全国大会,3K-5,1994.

付録

以下に実験.1 の実行例の一部を示す．

```
chmod = change mode
1 (0.67) chmod = change file mode
2 (0.50) bifchmod = change mode of a BIF file
3 (0.50) sdfchmod = change mode of an SDF file
4 (0.25) chfn = change finger entry
5 (0.25) passwd = change login password

cmp = compare two files
1 (1.00) cmp = compare two files
2 (0.60) sccsdiff = compare two versions of an SCCS file
3 (0.50) diff,diffh = differential file comparator
4 (0.33) diff3 = 3-way differential file comparison
5 (0.25) dircmp = directory comparison

comm = select or reject lines common to two sorted files
1 (1.00) comm = select or reject lines common to two sort
ed files
2 (0.27) cut = cut out selected fields of each line of a
file
3 (0.25) rev = reverse lines of a file
4 (0.25) cmp = compare two files
5 (0.22) uniq = report repeated lines in a file

cpp = the C language preprocessor
1 (1.00) cpp = the C language preprocessor
2 (0.25) cc = C compiler
3 (0.20) atrans = translate assembly language
4 (0.20) astrn = translate assembly language
5 (0.17) cflow = generate C flow graph

cp = copy
1 (0.33) rcp = remote file copy
2 (0.33) uucp,uulog,uuname = UNIX system to UNIX system c
opy
3 (0.25) bifcp = copy to or from BIF files
4 (0.25) cat = concatenate, copy, and print files
5 (0.25) lifcp = copy to or from LIF files
○6 (0.25) cp,ln,mv = copy, link or move files

date = print and set the date
1 (0.75) date = print or set the date and time
2 (0.33) slp = set the options for a non-serial printer
3 (0.29) hostname = set or print name of current host sys
tem
4 (0.25) cal = print calendar
5 (0.25) pr = print files

dc = desk calculator
```