文字列・単語列・文献検索ツールの実装

内山 将夫　　井佐原 均
通信総合研究所

概要

文字列検索のツールとしてTea，単語列の検索および統計量の計算のためのツールとしてSR，文献(情報)検索のためのツールとしてruby-irを実装した．本稿では，これらのツールの概要を述べる．これらのツールは，Linuxで稼働(Teaは Windows 98/2000/NTでも稼働)しており，以下より，Rubyと同じライセンスでダウンロードできる．

http://www.crl.go.jp/jt/a132/members/mutiyama/softwares.html

1 Tea

TeaはKWIC (KeyWord In Context) ツールである．TeaはSUFARY(山下 1999)を文字列検索エンジンに利用している．SUFARYはsuffix array (Manber and Myers 1990)を索引に利用することにより大規模テキストに対して高速に文字列検索できる．Teaでは，複数のsuffix arrayに対して順番に検索することにより，数ギガバイトのテキストが検索できる[1]．Teaは，現在，日本語(EUC-JP, Shift-JIS)と英語(ASCII)で動作している[2]．

TeaのGUI (Graphical User Interface)を，図1に示す．Teaは，図1に示されるように，いくつかの独立した領域からなる．まず，左側の領域は，木構造に組織化された用語集である．この用語集のノードをクリックすることにより，そのノードに関連付けられた用語を検索できる．たとえば，図1は，「温かい心遣い」を検索した例である．この用語集は，その場で対話的に編集したり，あるいは，XMLファイルとしてエディタ等により作成することができる．次に，右側の領域は，検索語入力部とKWIC表示部である．検索語入力部は上部にある．ここに検索したい文字列を入力すると，その検索結果がKWIC表示される．もし多量の検索結果がある場合には，それを一様サンプリングした結果が表示される．また，KWIC表示された各行をクリックすると，その文脈が下部に表示される．

KWIC表示された行は，各種の方法でソートできる．たとえば，最も基本的には，左右の文脈を辞書順にソートする．また，わかち書きされたテキストでは検索語との共起単語を共起頻度や共起強度(Dunning 1993)に応じてソートしたりできる．更に，日本語テキストで，わかち書きされていない場合でも，左右の文字n-gramの頻度によりソートできる．

2 SR

SR (Sequence Retrieval Library)は，100M単語程度からなるコーパスでの種々の統計量を計算するためのCライブラリであり，Rubyから使うことを意図して設計されている．

SRの機能としては，(1) 任意の単語列の検索，(2) 全ての n-gram の枚挙 (Nagao and Mori 1994)，(3) 全文類似検索 (Shang and Merrettal 1996)，(4)n-gram についての自己相互情報量 (Church and Hanks 1989)，対数尤度比 (Dunning 1993) 等の計算，等がある．

SR の Ruby モジュールを利用して作ったGUIを図2に示す．図2は，British National Corpusを検索しているときのものであるが，ツール自体は，コーパスが形態素に空白で区切られていれば日本語のコーパスにも適用できる．

[1] Teaでは，数メガバイト程度のテキストで，索引が不要なようなものについては，索引を利用せずに，単に，テキストの先頭からRubyのCライブラリを利用して正規表現検索をすることもできる．この場合には，Teaは，grepにGUI (Graphical User Interface)が付いたものと考えられる．TeaのGUIは，適切なメソッドを用意すれば，任意の検索エンジンから利用できる．

[2] GTK+を表示に使っているためその他の文字コードにも適用できると予想しているが，実際に試してはいない．

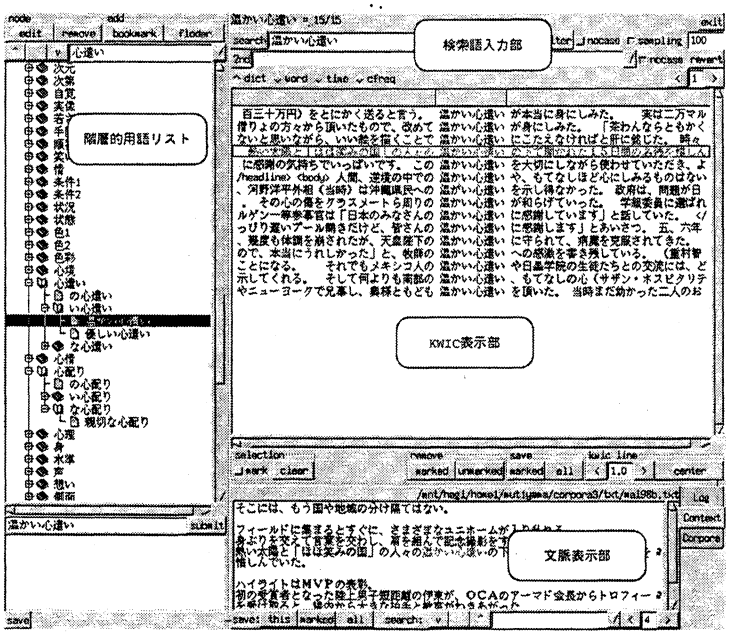

図 1: Tea の実行画面

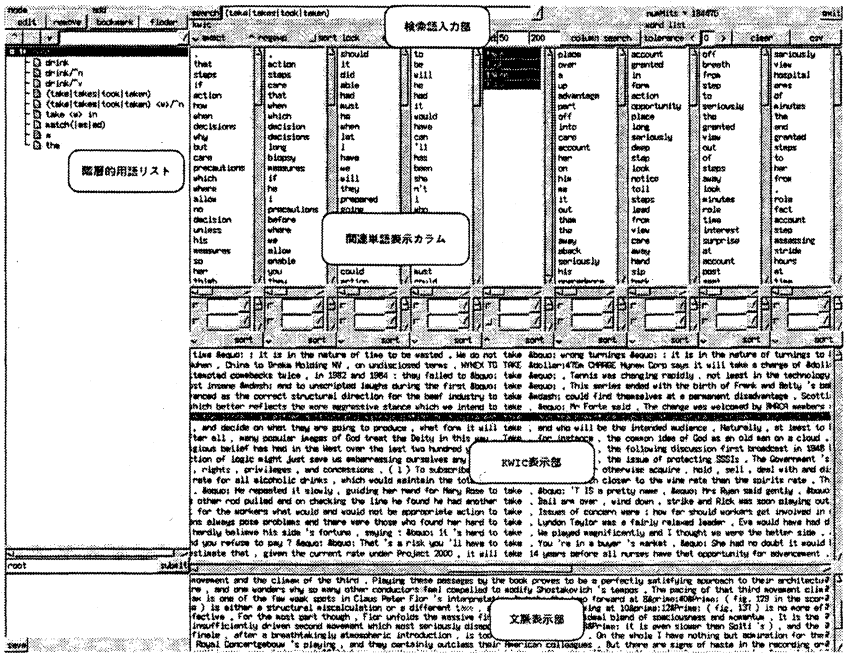

図 2: SR の実行画面

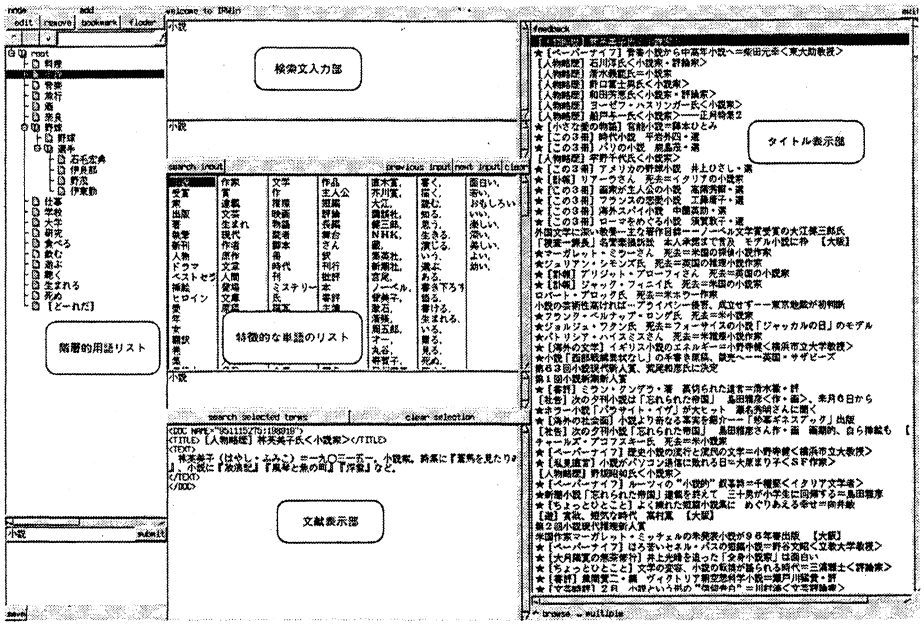

図 3: ruby-ir の実行画面

図 2では,「(take|takes|took|taken)」, つまり, これら 4 単語のいずれかにより検索している. その検索結果において, 各コラムには,「take」等との連関度 (対数尤度比) が高い単語がリストされている. ここで, 各コラムの位置は, コーパス中での「take」等との相対位置に対応している. たとえば「take」等の右隣りには「place, over, a, up, advantage」などがあるが, これは,「take place」などが顕著に出現していることを示す. また, 各コラムにある単語を選択することにより, 選択された単語からなる検索質問を作ることができる. たとえば,「take」等の右一つおきのコラムで「account」を選択したとすると,「(take|takes|took|taken) 任意の 1 単語 account」が検索される.

3 ruby-ir

ruby-ir (Information Retrieval Module for Ruby) は, 確率に基づくタームの重みを利用した情報検索手法 (Robertson and Walker 1994) を実装した Ruby モジュールであり, その検索精度は高い (内山 井佐原 2001). ruby-ir には, 日本語用と英語用のターム切り出しのためのモジュールが付属している[3]. そのため, 日本語と英語について情報検索をすることができるし, タームの切り出し用のモジュールを別途用意すれば, 他の言語についても情報検索をすることができる.

ruby-ir は, 元々は, バッチ的な処理のために作ったが, 情報検索では, ユーザとのインタラクションが重要である. なぜなら, ユーザは, 通常は, それほど明確な検索要求を持っているわけではないので, インタラクションを通じて, ユーザの検索要求を引出す必要があるからである. ruby-ir では, その目的のために, 質問文から検索された文献集合において特徴的な単語をリストすることにより連想検索を支援することを考えた[4].

図 3には,「小説」という単語を質問として

[3] 日本語には茶筅 (松本, 北内, 山下, 平野, 松田, 高岡, 浅原 2001) を用い, 英語には (Frakes and Baeza-Yates 1992) 付属のプログラムを用いている.

[4] 特徴単語を表示することにより連想検索を支援するシステムとしては (Takano 2001) がある.

入力したときに特徴的な単語として，普通名詞としては「小説，作家，文学，作品,...」，固有名詞としては「直木賞，芥川賞，大江，講談社,...]，動詞としては「書く，描く，読む，知る,...」形容詞としては「面白い，若い，おもしろい，いい，楽しい,...」がリストされている．このように単語をカテゴライズしてリストすることにより，単に特徴的な単語を並べたときよりも，素早く適当な単語を見付け出せることが期待できる．なお，ここでは，品詞により特徴的な単語をカテゴライズしているが，シソーラスなどを利用して意味素性を獲得し，それによりカテゴライズすることも考えられる．これらの単語は，クリックすることにより，検索質問に組込まれるので，ユーザは，最初の曖昧な質問から，インタラクションを通して，質問文を詳細にしながら検索をすることができる．

特徴的な単語の定義は，単語 w のスコアを $s(w)$ としたとき，$s(w)$ が上位にあるということである．ここで，$s(w)$ は，$\frac{f_{11}}{f_{1.}} <= \frac{f_{21}}{f_{2.}}$ のときには，$s(w) = 0$ であり，そうでないときには，対数尤度比 (Dunning 1993) に基づくものであり，以下の式で定義される[5]．

$$s(w) = 2\sum_{i,j} f_{ij} \left\{ \log \frac{f_{ij}}{F} - \log \frac{f_{i.} f_{.j}}{F^2} \right\} \qquad (1)$$

ただし，F はデータベースでの全文献数であり，

f_{11} = 検索された文献中で単語 w を含む文献数
f_{12} = 検索された文献数 $- f_{11}$
f_{21} = 全文献中で単語 w を含む文献数 $- f_{11}$
f_{22} = $F - f_{11} - f_{12} - f_{21}$

である．また，$f_{i.} = f_{i1} + f_{i2}$，$f_{.j} = f_{1j} + f_{2j}$ である．

[5](Hisamitsu and Niwa 2001) では，対数尤度比を含む複数の尺度について，特徴的な単語を選び出す尺度としての有効性を調べている．ここで，尺度構成上重要なこととして，単語の頻度の数え方，および，注目する文献集合の定め方がある．この 2 点において，本稿と (Hisamitsu and Niwa 2001) とは異なる．まず，単語の頻度の数え方であるが，(Hisamitsu and Niwa 2001) では tf (term frequency) を用いているが，本稿では，df (document frequency) を用いている．頻度として tf が良いか df が良いかは，今後，実験により確かめるべき事柄である．次に，文献集合の定め方であるが，(Hisamitsu and Niwa 2001) では，ある単語 v について，その単語 v が出現した文献集合を考え，その文献集合に基づいて尺度値を計算している．そのため，(Hisamitsu and Niwa 2001) では，単語 w について，ある単語 v との連関度としての尺度値を計算していることになる．一方，本稿では，検索された文献集合を考え，その文献集合に基づいて尺度値を計算している．そのため，本稿では，単語 w について，検索質問との連関度としての尺度値を計算していることになる．本稿の方法によると，質問文がたまたま 1 単語 v であった場合には，(Hisamitsu and Niwa 2001) と同じ文献集合を考えることになる．すなわち，本稿での文献集合は，(Hisamitsu and Niwa 2001) での文献集合を一般化したものとなっている．

参考文献

Church, K. W. and Hanks, P. (1989). "Word Association Norms, Mutual Information, and Lexicography." In *Proc. of ACL-89*, pp. 76–83.

Dunning, T. E. (1993). "Accurate methods for the statistics of surprise and coincidence." *Computational Linguistics*, **19** (1), 61–74.

Frakes, W. B. and Baeza-Yates, R. (Eds.) (1992). *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall.

Hisamitsu, T. and Niwa, Y. (2001). "Topic-Word Selection Based on Combinatorial Probability." In *NLPRS-2001*, pp. 289–296.

Manber, U. and Myers, G. (1990). "Suffix arrays: A new method for on-line string searches." In *Proc. of the First ACM-SIAM Symposium on Discrete Algorithms*, pp. 319–327.

Nagao, M. and Mori, S. (1994). "A New Method of N-gram Statistics for Large Number of n and Automatic Extraction of Words and Phrases from Large Text Data of Japanese." In *Proc. of COLING'94*, pp. 611–615.

Robertson, S. E. and Walker, S. (1994). "Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval." In *SIGIR-94*, pp. 232–241.

Shang, H. and Merrettal, T. H. (1996). "Tries for Approximate String Matching." *IEEE transactions on Knowledge and Data Engineering*, **8** (4), 540–547.

Takano, A. (2001). "Associative information access using DualNAVI." In *NLPRS-2001*, pp. 771–772.

内山将夫 井佐原均 (2001). "情報検索パッケージの実装." 情報処理学会研究報告 情報学基礎 63-8 (2001-FI-63).

山下達雄 (1999). "SUFARY ガイド." http://cl.aist-nara.ac.jp/lab/nlt/ss/.

松本裕治，北内啓，山下達雄，平野善隆，松田寛，高岡一馬，浅原正幸 (2001). ''形態素解析システム 『茶筌』 version 2.2.5 使用説明書.'' http://chasen.aist-nara.ac.jp/.