小学館コーパスクエリーシステム–構造化文書の検索インタフェースと照合技術について

EST コンサルティング本部　中 村 隆 宏

1．概要

私は、1996年より株式会社小学館殿（以下、小学館）のマルチメディア局に、常駐で勤務している。現在の主な業務はコンサルティングと業務に特化した研究である。この度、小学館コーパス*1クエリーシステム（以下、小学館 CQS）の開発に携わったので、その詳細について紹介する。

小学館 CQS の開発において、中心的なテーマとなった構造化文書の検索インタフェースと照合手法について述べる。本システムが扱ったデータは、特定の SGML*2（Standard Generalized Markup Language）文書であるが、今後普及すると考えられる XML*3（eXtensible Markup Language）文書などの構造化文書一般への高度な情報検索に応用できる技術である。

2．背景

標準化された構造化文書の普及は、インターネットの普及によって HTML から始まった。ユーザー指向の詳細な構造定義を可能にする SGML や XML の普及も、その裾野の上に広がっている。インターネットの普及は、同時に、その膨大なテキストの検索サービスを実現させるために検索エンジンの普及をもたらした。構造化文書をデータベースのレコードのように扱うメリットが知られると、構造化文書を扱えるデータベースエンジンの必要性が認識されるに至った。

一方、インターネットの普及とは無関係に、構造化文書と全文検索エンジンの開発が足並みを揃えて行われてきた場所がある。専門領域で扱われる特定のテキストデータは、すでに80年代から SGML 文書に変換され、エンコードの指針を与えるべく、タグセットの標準化を検討するコンソーシアムが運営されてきた。これは、産業応用ではなく学術や出版から始まった。辞典開発において、「Oxford English Dictionary 第二版」の改訂のために、初版本がすべて SGML 化され、改訂作業を円滑に行うために専用の検索エンジンとして OpenText が開発された。OpenText は、SGML ライクな文書のための全文検索エンジンと捉えてもよく、最大の特徴は、任意の要素名をキーにして、そのタグ終了までの範囲に限定して検索できることである。このエンジンは、本プロジェクトでも活用しており、詳細説明を後述する。

また、各国の古典の文学作品などを集めたテキストアーカイブの開発に合わせて、エンコードの指針を協議するコンソーシアムが運営され、ジャンルごとに最適な DTD*4（Document Type Definition）が開発されてきた。本プロ

＊1）コーパス（Corpus）：自然言語処理および言語学、さらに言語教育の実用の場において使用する言語情報資源。単なる文の集りではなく、構文構造などの付加情報をタグとして付加したコーパスは、研究上も重要な言語情報資源であるため、タグの付け方自体も自然言語処理や文書処理の大きな分野になりつつある。

＊2）SGML：文書の論理構造、意味構造を簡単なマークで記述する言語。マーク付けすることで、後に抄録や索引を作るなど、文書をデータベースとして利用できるようになる。

＊3）XML：標準化が進んでいるページ記述言語。HTML で普及したリンク機能を拡張し、SGML をインターネット向けに最適化するなど、それぞれの長所を併せ持っている。

＊4）DTD：マークアップの構成要素、その順序（親子関係）、指定できる属性などについて定めた規則。

ジェクトで紹介するコーパスと呼ばれる用例データベースも、そのガイドラインに沿ってエンコードされている。

3．コーパスの遷移とBNC

3.1　コーパスに基づく辞典開発

小学館では、語学辞典の開発や改訂の元になるデータベースのインフラ構築を行ってきている。インフラの構成要素は2つある。1つは、既存のデータ資産のデータベース化である。これは各種の電子辞典プロジェクトや書籍改訂の際に，SGML文書化を行っている。もう1つは、コーパスと呼ばれる言語資源を背景にした言語研究の環境提供や辞典開発という、まったく新しいシステム構築である。

英国では80年代後半から、コーパスに準拠した英語辞典開発が、複数の出版社、大学、政府を巻き込んで行われてきた。残念なことに、そこで構築されたコーパスには、一部の研究者が研究用途にアクセスできるのみで、国外の出版社が商用に利用することができなかった。また、アクセスできても、非常に貧弱な検索インタフェースしか用意されていなかった。しかし、2001年にBNC（British National Corpus）という総語数1億語のコーパスが、欧州圏以外にもCD-ROMで一般にリリースされたことで、事情が一変する。小学館でも、これをすぐに入手して、データベース化と検索要件の洗い出しに入った。また、アメリカ国内でも、BNCをモデルにしたコーパス構築のプロジェクトANC（American National Corpus）が開始され、国内の辞典出版社のすべてが出資して協賛している。

3.2　BNC

BNCを例に取りながら、SGML文書としては、やや特殊な構造をもつコーパスのデータ形式について説明しておく。これは、後述する検索の仕様と密な関係にある。

BNCのコーパスデータは、約4000のテキストファイルから構成される。ファイルサイズで約1.5GBである。収録されたコンテンツは、多様なジャンルからバランスよく収集され、全体として現代の英国の英語を代表する用例集となるように配慮されている。語の頻度や、ジャンルにわたる頻度の比較を行うことを、あらかじめ配慮して、1ファイルの長さを、ある程度まで揃える努力がされている。また、全体の1割は、実際の講演の模様や協力者の日常会話を収録して書き起こしたテキストが占めている。収録の場所も全国にわたっており、年齢層、性別の分布にも統計的な配慮がなされている。BNCに興味がある方は公式ホームページ[*5]を参照されたい。ファイルは、書誌的なデータやジャンルなどの詳細な分類情報を収めたヘッダー部と本文からなる。この構成は一般的だが、本文部の構造に最大の特徴がある。本文データは、“文”という要素タグで一文ごとに管理される。また、文を構成する語は、“語”という要素タグで、一語ごとに管理される。文には、通しの文番号や語数が属性で記述される。文が発話の場合は、話者IDが属性として付随し、ヘッダー部を参照することで、話者の性別、年齢、出身地などの情報を得ることができるようにコーディングされている。語のタグには、その語の文法的な属性として、名詞や動詞を表わす品詞記号が割付けられている。配布されているBNCのオリジナルデータには、品詞記号しか付いていないが、種々の検索では、文中での活用語尾の変化は無視したいので、小学館の英和辞典を利用して、後日に原形表記のデータを属性として付加してある。

2種のソースデータを下記に示す。BNCオリジナル形式は、配布されたままのSGMLデータである。インデキシング形式では、検索に適したインデキシングが行えるようにデータ構造を変えている。具体的には、語とその属性をばらして、順序関係を保存したまま、別のタグフィールドにまとめている。タスクとしては、語と属性の組み合わせよりも、連続する語や属性のパターンが多いと想定されたからである。

＜BNCオリジナル形式＞

＜s n＝“20”＞＜w NN1＞Painting ＜w NN2＞holidays ＜w VVB＞offer ＜w AT0＞an ＜w AJ0＞ideal ＜w NN1＞chance ＜w TO0＞to ＜w VVI＞relax ＜w PRP-AVP＞in ＜w AT0＞a ＜w AJ0＞different ＜w NN1＞environment ＜w CJC＞and ＜w VVB＞improve ＜w DPS＞your ＜w NN1＞painting＜c PUN＞, ＜w CJC＞or ＜w VVB＞try ＜w AT0＞a ＜w AJ0＞new ＜w NN1＞subject ＜w CJC＞or ＜w NN1－AJ0＞medium ＜w PRP＞in ＜w AT0＞the ＜w NN1＞company ＜w PRF＞of ＜w AJ0＞other ＜w NN2＞students＜c PUN＞, ＜w PRP＞under ＜w AT0＞the ＜w NN1＞guidance ＜w PRF＞of ＜w AT0＞an ＜w NN1-AJ0＞expert ＜w NN1＞tutor＜c PUN＞.

＜インデキシング形式＞

＜s f＝“CL0”n＝“20”＞＜s0＞Painting holidays offer an ideal chance to relax in a different environment and improve your painting, or try a new subject or medium in the company of other students, under the guidance of an expert tutor.＜/s0＞＜w＞ ＜P＞ NN1 NN2 VVB AT0 AJ0 NN1 TO0 VVI PRP_AVP AT0 AJ0 NN1 CJC VVB

＊5）BNC公式ホームページ：http://info.ox.ac.uk/bnc/index.html

DPS NN1 c CJC VVB AT0 AJ0 NN1 CJC NN1_AJ0 PRP AT0 NN1 PRF AJ0 NN2 c PRP AT0 NN1 PRF AT0 NN1_AJ0 NN1 c ＜/P＞＜L＞ paint holiday offer a ideal chance to relax in a different environment and improve your paint, or try a new subject or medium in the company of other student, under the guidance of a expert tutor.＜/L＞＜/w＞＜/s＞

3.3　二次情報付文書データ

一般に、本文の他に品詞などの二次情報がついたコーパスを「注釈つきコーパス（Annotated Corpus)」と呼んで、本文からのみ成る従来のコーパスと区別している。コーパスに基づいて言語研究が詳細化、高度化するためには、二次情報の必要性が高まっており、またその二次情報をキーにした検索が必要となってきている。本開発では、こうした将来のニーズに応えることが、最大の課題であった。

4．CQS による検索の特長

4.1　検索の要求仕様

検索の実行には、以下の 5 つの要件を満たすことが必須である。

①ヘッダー情報を検索キーにできること
②属性を検索キーにできること
③キーとキーの間に任意の語数を許すワイルドカードが書けること
④OR 条件の指定ができること
⑤上記を簡潔に表現できる構文とそのインタープリタを開発すること

こうした検索の要件を満たす CQL（Corpus Query Language）の言語仕様を、まず策定した。CQL をユーザーに直接書かせることはできないので、検索キーの適切な投入だけで済む検索インタフェースを新規に開発した。これは、語の並びと、語に付属する属性の並びを直感的に扱えるようなテーブルとなっている。セルに投入できるキーとしては、通常の単語はもちろんだが、前方一致や OR 条件が書けるようになっている（図 1 参照)。

要は、BNC が持っている情報と構造を可能な範囲で利用できる検索条件が書けることが、このインタフェースの骨子となっている。なお、通常の全文検索と決定的に異なることは、複数のキー指定は、集合論的に扱われるのではなく、その前後関係、つまり文法を配慮したものになっていることである。2 語以上からなる複合語やフレーズの検索が、重要なタスクとなるため、検索キーの並びには順序関係が成立する。一方、属性の並びは、順不同である。

属性条件の指定について具体的に説明する。たとえば、表記形“look”は、それだけでは動詞か名詞か不明なので、名詞としての“look”だけを調べたい場合がある。「“look”かつ名詞」という 2 つの条件がキーとなるので、テーブル上では、図 2 のような検索キーとなる。

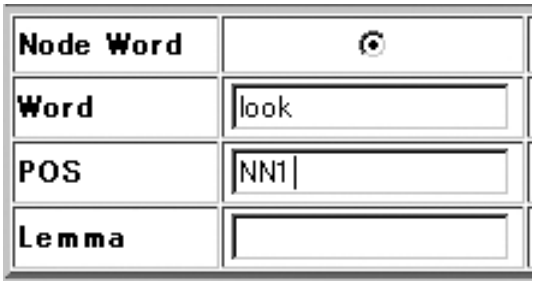


図 2　「“look”かつ名詞」の検索

あるいは、動詞 look は、主語の人称や単数複数によって、実際の文では、“looks, looked, looking”のように語尾変化を生じているので、“look”だけでは動詞の用法を全て検索できない。このために、先に述べた原形表記を利用して検索を行う。この例は、図 3 のようになる。

イメージとしては、検索 GUI と同じように、二次元配列のデータが文を単位として延々と並んでいる巻紙を想像していただきたい。論理的には検索 GUI で指定されたキーの二次元パターンで、パターンマッチングを行うことである。一般化すれば、「二次元配列データを、二次元パターンから自由に検索する」という特殊なデータ検索となる。もちろん、このパターンマッチングを高速に行うインデックスファイルの作成方法は一般に知られていないし、こうしたデータ構造に特化した検索エンジンは、知る限りにお

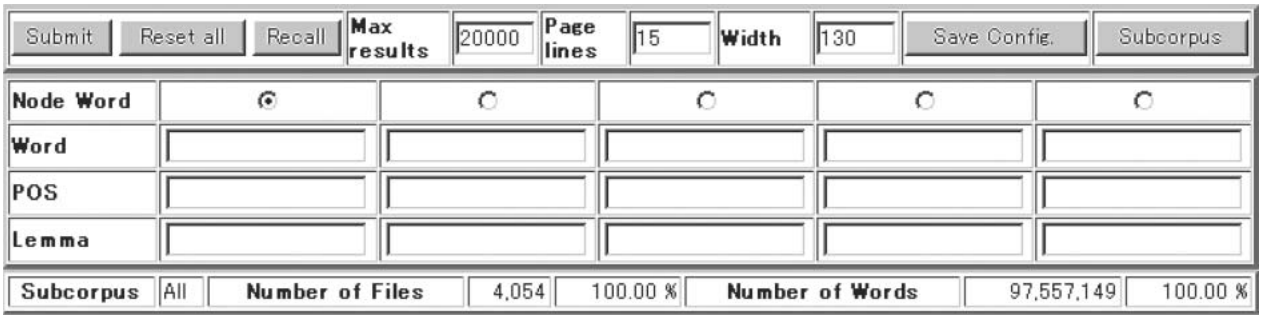


図 1　CQL の検索インタフェース

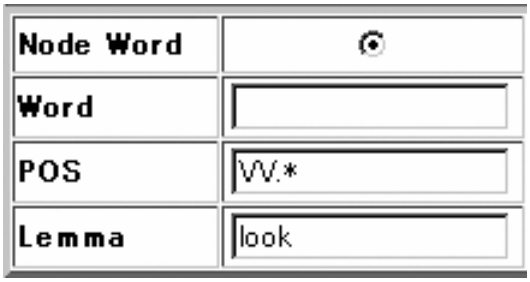


図 3 「“look”かつ動詞」の検索

いては未だに実用化されていない。

小学館 CQS の基本的なインタフェースは、図 4 のようにフレームで 3 分割に仕切られ、上段が検索キーのテーブル、中段が KWIC、下段がクラスター表示画面となっている。KWIC とクラスターについては、後述する。

4.2 サブコーパス条件

なお、コンテンツの書誌データをキーに指定する典型例として、サブコーパスの指定がある。これは、BNC 全体をヘッダー情報をもとにいくつかに仕切って、検索対象の範囲を限定して調査する場合に使用される。典型例は、ある語の使用頻度を会話文と書き言葉の集合で比較する場合である。性差、年齢差、地域差といった社会学的な視点からの語彙調査も多いようである。このように、サブコーパスを指定するキーも多岐にわたり、コマンド指定では使いづらいために、GUI から CQL に落としている。サブコーパス指定の GUI を図 5 に示す。

4.3 アプリケーション層

検索エンジンを新規に開発することは、種々の制約から不可能であるため、すでに使い慣れた OpenText を活用するという方法を採った。

CQL は、GUI から受け取った各種キーのパラメータから一旦作成されて、CQL インタープリタに渡される。CQL インタープリタは、検索エンジンの OpenText に渡す PAT コマンドを生成し、その検索結果への二次検索のための検索パターン式を正規表現にて作成する。検索効率が最大となる PAT コマンドと二次検索用の正規表現パターンの生成が、ここでの最大の要件となる。各ソフトウェアモジュールの概略の関係は、図 6 のようになる。

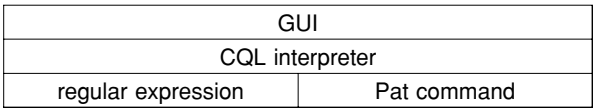


図 6 各ソフトウェアモジュールの関係

見出し語“take”の直後に副詞が来る用例を検索するときに使用する CQL、PAT コマンド、正規表現（RE）を順に示す。

CQL：* {L=take} {P＝“AV.＊”}

PAT：((region s including(region L including“take”))

ˆ ((((region s including (region P including

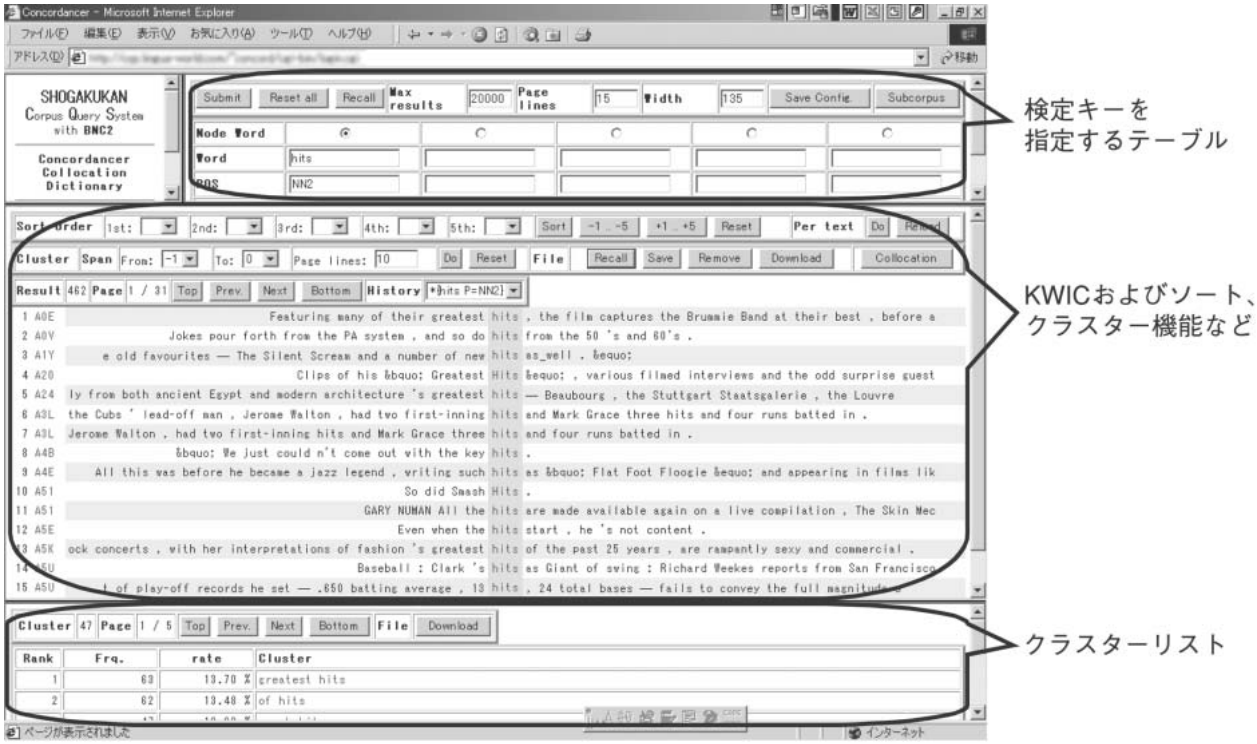


図 4 小学館 CQS のインタフェース

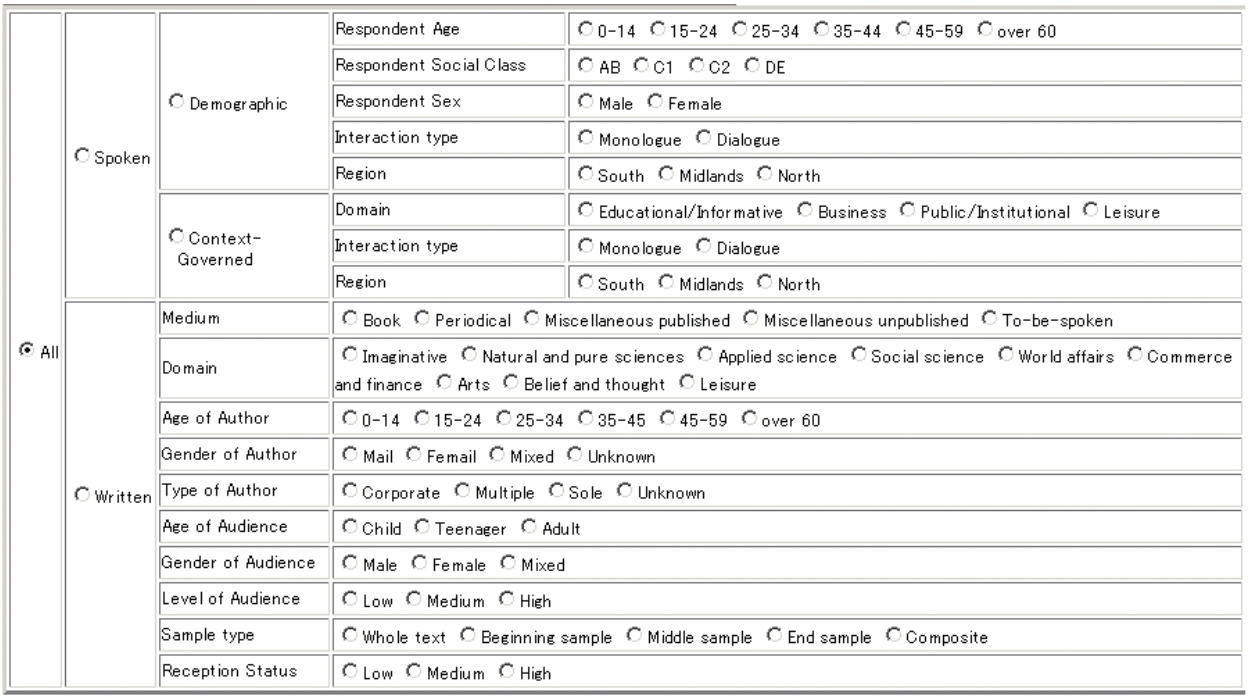


図5　サブコーパス指定の GUI

“AV”))))))

RE:()(¥s*＜w¥s＋P¥s*＝¥s*¥w＋¥s＋L¥s*＝¥s*take¥s*＞¥s*[¥w¥－¥']＋)(¥s*＜w¥s＋P¥s*＝¥s*(？:AV¥w*)¥s＋L¥s*＝¥s*¥w＋¥s*＞¥s*[¥w¥－¥']＋)

(¥s...スペース ¥w...単語に使われる文字クラス)

CQL インタープリタは、シェルコマンドとして整備してあり、バッチジョブを流して大量の検索ができるようになっている。

4.4　OpenText

冒頭でも触れたが、OpenText について再度、説明しておく。この全文検索エンジンは、ウオータールー大学（カナダ)、英国オックスフォード大学出版局、およびカナダ政府の共同事業として進められた OED（Oxford English Dictionary 第二版）の電子化プロジェクトにおいて、1980年代の後半から同大学の Tim Bray を中心に開発された。Tim Bray はその後、W 3 C や XML の分野で活躍して一般にも知られるようになり、現在は Textuality 社の代表となっている。なお、同社の業務内容は、XML のコンサルタントである。

OpenText は、インデックスファイルにパトリシアツリー方式*6（Patricia : Practical Algorithm To Retrieve Information Code In Alphanumeric）と呼ばれるデータ形式を用いて、いかなる部分文字列についても、その開始位置を記したファイルオフセット値の集合を高速に返すのが特徴である。しかし、本当の機能特徴は、SGML 文書の構造を反映した検索ができることである。これは、非常に階層が深くなる DTD をもつ OED のような語学辞典においては、必須の機能として要求されたのだと容易に想像できる。

次期バージョンの OpenText では、Unicode 対応、XML の属性を検索キーにできること、インデックスファイルの 2 GB 制限の撤廃などが予定されている。

4.5　検索の戦略

OpenText は、単純な文字列の検索は非常に高速であり、また特定タグ内に範囲を限定した検索も得意である。しかし、本件のように二次元配列へのパターンアクセスが直接にできるわけではない。あくまでも、OpenText が返す一次データを可能な限り最小の量に抑えて、あとは愚直に逐次的に文字列の照合を繰り返すという戦略しか現状では方法がない。これは極端な場合、OpenText が BNC の全量を返した場合は、最初から BNC 全体を照合する手間に相当する。ただし実際には、そういう検索のタスクは稀である。

OpenText は、インデキシング時に検索キーにしたい要

＊6）パトリシアツリー方式：検索キーの文字列が、どの半無限部分文字列と一致するかを判定して検索する方式。文字列を単なるビットストリームと見なし、ビット判定を基本にした二分木により照合の判定（探索）を行う。このために、重複して 2 度以上通るパスがないために無駄が無い。

素タグ名を指定ファイルに列記しておくと、本文のインデックスデータとは別に、その要素の開始から終了までのインデックスを別途作成する。特定のタグの範囲に含まれる複数キーの検索で、元来ヒット件数が多いものを AND 条件に書くと、期待しているパフォーマンスが出ない。集合演算の場合、少ない母集団から照合を行うことで改善されはずだが、項の前後関係を変えても、実際には改善されなかった。これはベンダーの弁によれば、パトリシアツリー自体が集合演算に向いていないからということである。

CQL 仕様案では、検索の結果同士の集合演算を盛り込んである。しかし、実際のシステムでは論理和だけを後処理にて実装している。これは、検索結果をサーバーのユーザー・ディレクトリに残し、後で KWIC 画面に呼び出す機能を利用して、複数の結果を同じ画面に呼び出せるようにしたものである。1 回の検索式では書けないパターンを 2 回に分けて検索する場合などにも重宝している。

KWIC は“Key Word In Context”の略で、文書検索の結果表示の形式である。検索語がリストの中央の列に並ぶように検索結果を配置するもので、検索語の前後関係を上下行で比較したりする場合に重宝する。なお、検索語を挟む前後の語の位置にソートキーを設定して行を並べなおすことで、検索語を含む特定の表現を調査できる。コーパスの検索ソフトウェアが、最低限に備えているインタフェースである。

4.6 パフォーマンス

以下に、小学館 CQS と egrep での比較を行ったデータを示す。まず、タスク、CQL、egrep のコマンドと検索時間、そして小学館 CQS での検索時間と並べる。タスクは 3 例を挙げている。なお、小学館 CQS の正規表現検索は Perl を用いているため、その部分の照合スピードは元来、grep に及ばない。なお、正規表現中に見られる記号類の説明は省略する。

タスク「名詞の“hits”を含む用例を探せ」
CQL = {W = hits&P = NN2}
◇grep
コマンド：egrep '＜w NN［＾＞］＋＞hits［＜]'
時間　　：2：36（2 分36秒　以下同様）
◇CQS
時間　　：0：18

タスク「動詞＋前置詞もしくは副詞のあとに、最大 2 語を挟んで名詞の“corner”がくる用例を探せ」
CQL = {P = VV. *, P = (PRP｜AV. *), [0, 2], L = “corner”&P=NN. *}
◇grep
コマンド：egrep '＜w VV[＾＞]＋＞[＾＜]＋＜w(PRP｜AV[＾＞]＋)＞[＾＜]＋(＜[＾＞]＋＞[＾＜]＋)*＜w NN[＾＞]＋＞(corner｜corners)'
時間　　：2：48
◇CQS
時間　　：1：44

タスク「give の全ての活用形＋“up”＋冠詞類のあとに最大 1 語を含んで名詞もしくは現在分詞がくる用例を探せ」
CQL={W = (give｜gives｜gave｜given), W = “up”, P = (DT0｜AT0｜DPS), [0, 1], P = (NN. *｜VVG)}
◇grep
コマンド：egrep '＞(give｜gives｜gave｜given)＜w[＾＞]＋＞up ＜w(DT0｜AT0｜DPS)＞[＾＜]＋(＜[＾＞]＋＞[＾＜]＋)？＜w(NN[＾＞]＋｜VVG)＞'
時間　　：2：48
◇CQS
時間　　：2：30

これらの結果は、よりありふれた語や品詞を、より多く OR 条件などで結んだ場合には、単純な文字列検索との差が出にくいということを表している。タスクとしては、それほど特殊なものではないため、今後は改善しなければならない。改善のポイントは、最適な PAT コマンドの生成と、タスクに最適化されたインデックスソースデータのエンコード形式を検討するという 2 点である。既に、改善の方針は立ててあるので、予定が立てば実装したい。

最適化のためのチューニングメニューとは別に、検索エンジンの新規開発のためのアイデアも構想中である。

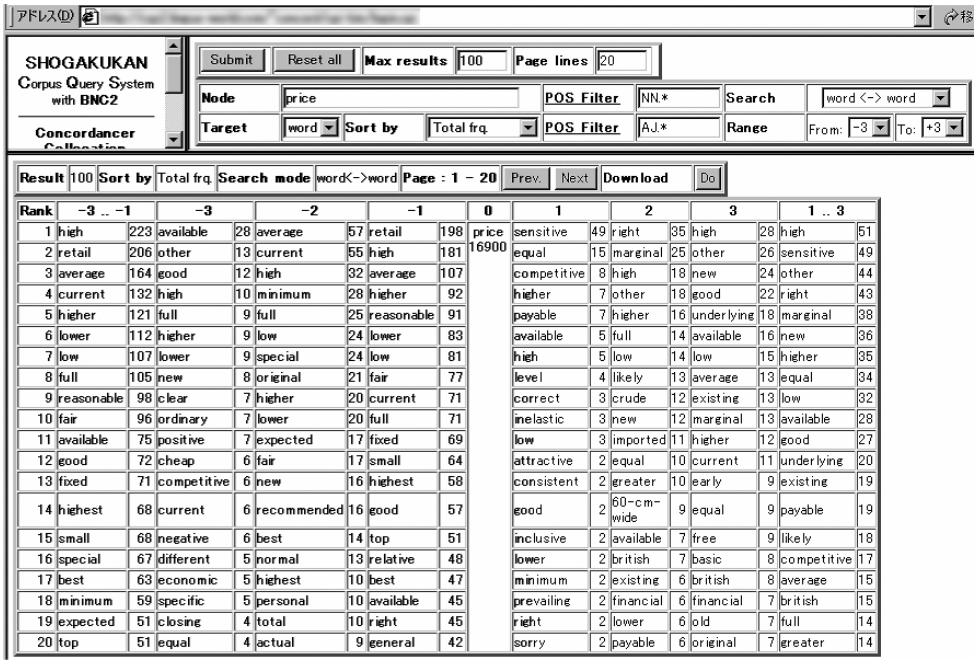
5．共起

5.1 二語間の共起検索

「共起（Collocation）」とは、語と語が近傍に、同時に現れることを指す言語学上の専門用語である。共起の頻度データは、BNC のような大規模な言語データから収集することによって、語の用法に重要な知見が得られるため、コーパスの検索システムでは重視される機能である。

小学館 CQS では、二語間の共起については、すべての BNC データから事前にデータベースを構築してサービスできるようにしてある。図 7 は検索結果を示す共起テーブルである。“price”という名詞の前後にどういう形容詞が共起するのか、総頻度で各位置ごとに集計したテーブルである。形容詞の例だけを取り出せるのは、元来の BNC データに品詞情報が付いている恩恵に他ならない。もちろん、ユーザーは見たい品詞だけを選択することができる。

共起テーブルのツールとしての限界は、あくまでも二語



Rank	−3 .. −1		−3		−2		−1		0	1		2		3		1 .. 3	
1	high	223	available	28	average	57	retail	198	price 16900	sensitive	49	right	35	high	28	high	51
2	retail	206	other	13	current	55	high	181		equal	15	marginal	25	other	26	sensitive	49
3	average	164	good	12	high	32	average	107		competitive	8	high	18	new	24	other	44
4	current	132	high	10	minimum	28	higher	92		higher	7	other	18	good	22	right	43
5	higher	121	full	9	full	25	reasonable	91		payable	7	higher	16	underlying	18	marginal	38
6	lower	112	higher	9	low	24	lower	83		available	5	full	14	available	16	new	36
7	low	107	lower	9	special	24	low	81		high	5	low	14	low	15	higher	35
8	full	105	new	8	original	21	fair	77		level	4	likely	13	average	13	equal	34
9	reasonable	98	clear	7	higher	20	current	71		correct	3	crude	12	existing	13	low	32
10	fair	96	ordinary	7	lower	20	full	71		inelastic	3	new	12	marginal	13	available	28
11	available	75	positive	7	expected	17	fixed	69		low	3	imported	11	higher	12	good	27
12	good	72	cheap	6	fair	17	small	64		attractive	2	equal	10	current	11	underlying	20
13	fixed	71	competitive	6	new	16	highest	58		consistent	2	greater	10	early	9	existing	19
14	highest	68	current	6	recommended	16	good	57		good	2	60-cm-wide	9	equal	9	payable	19
15	small	68	negative	6	best	14	top	51		inclusive	2	available	7	free	9	likely	18
16	special	67	different	5	normal	13	relative	48		lower	2	british	7	basic	8	competitive	17
17	best	63	economic	5	highest	10	best	47		minimum	2	existing	6	british	8	average	15
18	minimum	59	specific	5	personal	10	available	45		prevailing	2	financial	6	financial	7	british	15
19	expected	51	closing	4	total	10	right	45		right	2	lower	6	old	7	full	14
20	top	51	equal	4	actual	9	general	42		sorry	2	payable	6	original	7	greater	14

図7　小学館 CQS の共起テーブル

の間の関係しか問えないということである。実際には、“give up”のような句動詞（Phrasal Verb）と共起しやすい名詞が知りたいというニーズは高い。小学館 CQS がこうした要求にも応えられることを次に示す。

5.2　句による共起検索

例えば、「句動詞“look forward to”が目的語に取る名詞や動名詞の頻度リストを調べたい」という検索式は、図8のようになる。なお、“to”の直後に希望する名詞や動名詞が来るとは限らず、冠詞や形容詞などが挿入される可能性がある。そのため、その部分をワイルドカードにするために挿入語の最大語数を指定している。さらに、KWIC 画面の中央に配置させる語（ノードワード）を、通常なら“look”にするところを、品詞の変数の個所に指定する。こうすると、KWIC の中央には、調べたい名詞や動名詞が並ぶ。なお、このようにノードワードの位置を変数の位置に指定できる機能をもった同種のソフトウェアは他にない（図9参照）。

KWIC には、語の任意の長さ（スパン）にわたってクラスターを集計して頻度順に並べる機能がある。この機能を使用してノードワードの集計が簡単にできる。図10はその結果である。

6．まとめ

小学館 CQS で実現された検索システムについて述べた。コーパスという二次情報付文書への複雑な問い合わせを、簡単明瞭なインタフェースで実現していることを理解いただけたら幸いである。

他の分野への応用であるが、学術論文の検索サービスを目的に、専用の検索エンジンを新規に開発する研究プロジェクトが、国立情報学研究所で行われている。既に、大量で複雑多岐な構造を持つ文書を抱え、高度な情報サービスを考えている場所でしか、こうしたシーズは成立しないものである。同じように、小学館殿が、BNC などのコーパスデータに早くから注目し、その検索システムの重要性と必要性をご理解いただけたことに深謝します。

XML の柔軟さは、ユーザーのアノテーション（二次情報）を容易にマスターデータへリンクさせる機会を増大させる。そうして深く構造化された広大な文書空間を、スムーズに往来できる検索エンジンの必要性は、今後ますます高まるだろう。

なお、昨2001年に韓国の延世大学で開催された、第2回アジア辞書学会にて発表した2件のペーパーのうち、以前の『SOFTECHS』に掲載した「インターネットと自然言語処理による新語情報システム」に関する論文*7が、優秀論文として同大学から出版されている『Studies in Lexicography』に再録された。この場を借りて報告する。

＊7）『SOFTECHS』Vol. 23, No. 2（2000年11月発行）掲載の「インターネットと自然言語処理による新語情報システム」。

Node Word	○	○	○	○	◉
Word		forward	to	[0,2]	
POS	VV.*				NN.*|VVG
Lemma	look				

図 8　「句動詞“look forward to”が目的語に取る名詞や動名詞の頻度リストを調べたい」という検索

Result 1586 Page 1 / 80 Top Prev. Next Bottom History [P="VV.*" L=look] forward to [0,2] *[P="(NN.*|VVG)"]

1 A00	month contract with ACET , is looking forward to caring for	people	with HIV/AIDS .
2 A04	g of many types may assist a reader who looks forward to an	encounter	with a work of art .
3 A0D	&bquo; I shall look forward to Saturday	evening	then . &equo;
4 A0E	iences as much as audiences need films , we look forward to	seeing	you .
5 A0U	I had been quite looking forward to	sampling	its unsavoury delights .
6 A0Y	The supporters at Devonshire Park are looking forward to	seeing	how Pam handles the transfer from the U.21 event to the seni
7 A17	We look forward to	seeing	you there .
8 A17	Marjorie was looking forward to her	freedom	; no need to be up at seven o'clock to go to work , no rushi
9 A1J	&bquo; We look forward to the	discussions	.
10 A22	Now I 'm looking forward to reviewing the	situation	after the Portsmouth match .
11 A23	We 're looking forward to the	opportunity	of helping . &equo;
12 A2E	we lost and for that reason I have been looking forward to	playing	in this event again . &equo;
13 A2H	d a good forward sales position can look forward to another	year	of growth . &equo;
14 A2P	lo-Irish agreement , which looked forward to increasing the	number	of Catholics in the RUC but refrained from mentioning the UD
15 A2S	orld champions , Price is merely looking forward to another	tussle	with the ultimate rugby enemy .
16 A2S	I 'm looking forward to the	match	and to seeing him training .
17 A3W	&bquo; We shall look forward to seeing the	report	and if there is new data that warrants investigation . &equo
18 A40	zal last Saturday from his mind , and is looking forward to	tackling	Michael Stoute 's colt again in the Breeders ' Cup Mile at G
19 A4J	of a bow , said : &bquo; Mr Billington , I look forward to	reading	you tomorrow and seeing what I think of the play . &equo;
20 A4U	Gardens today who are actually looking forward to Nigel 's	speech	.

図 9　ノーワードを変数の位置に指定できる機能

Rank	Frq.	rate	Cluster
1	115	7.25 %	seeing
2	56	3.53 %	day
3	53	3.34 %	meeting
4	36	2.27 %	going
5	29	1.83 %	future
6	28	1.77 %	time
7	24	1.51 %	working
8	24	1.51 %	getting
9	24	1.51 %	year
10	20	1.26 %	visit

図10　ノーワードの集計結果