Ｋａｐｐａ上での用例検索ツールの試作

奥西 稔幸　三吉 秀夫　阿部 ひろみ　小渕 保司
シャープ株式会社

1　はじめに

ユーザが指定したキーワードを満たす用例を文章データベースから検索するツールを開発中である[1]。フルテキストサーチのように単なる表層の文字列だけでなく、文法情報も含んだ複雑なキーワード（構造化キーワード）を指定できるため、辞書や文法を記述する際の実証データとなる言語現象を柔軟に検索することが可能である。現在、本ツールは逐次型推論マシンPSIの論理型言語ESP上で稼働している（用例検索ツールLTB-KWIC[2]）が、将来の大容量化の際に考えられる保守・検索効率での課題に対処するために、ICOTで開発中の知識ベース/データベース管理ソフトウエア（Kappa[3]）への移植を試みた。本稿では用例検索システムのKappaへの移植概要および、LTB-KWIC版との比較を中心にした評価を報告する。

2　用例検索ツールLTB-KWICとその問題点

LTB-KWICは、再検索、サンプル検索などに加え、結果同士の集合演算、ソート出力、センタリング表示、キーワード管理などの機能も備えている。マウスとメニューを中心にしたユーザフレンドリーなインタフェースにより、計算機に不慣れな言語学者でも容易に操作できる。検索の対象となるデータベースは形態素切りと文法素性付与が終った文章で、現在のところ「品詞,出現形,代表形,種類,活用型,活用行,活用形,意味」の８つの文法素性を設定している。例えば、図１のキーワードは『（人間）が（場所）で～する』という格パターンを表すが、

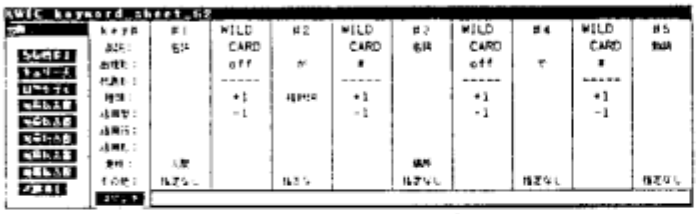

図１：キーワードの例

「（人間）が」と「（場所）で」と「～する」の間にワイルドカード（＊）を指定したり、「動詞」の「活用形」を任意にしているため、多くの用例が得ることができる。図１のキーワードによる検索結果を図２に示す。

図２：検索結果

現在、報告書や講演録を中心に選んだ15出典（約4000文）を提供しているが、今後さらにデータベースを追加した場合、保守・検索効率の面で次の２つが課題となる。

- 検索効率を上げるためのインデックスをESPクラスとして実現しているため、データベースの一部を追加・削除・修正するたびにデータベース全てを再コンパイルしてインデックスを生成する必要がある。
- ほとんど全ての文に表れるキーワードの場合、インデックスが有効に働かない。（例えば「格助＊動」）

3　知識ベース/データベース管理システムKappa

Kappa はICOTで開発中の知識ベース/データベース管理機能を提供するシステムで、今回の移植対象である第２版（Kappa-II）は以下の特徴を有する。

- 非正規関係モデルの採用
- 組識別子を用いた集合(中間関係)の実現
- ユーザ定義コマンドの登録と実行
- 主記憶データベース機能の実現

現在、並列検索版（Kappa-P）の開発が進んでいる。

4　Kappaへの移植

4．1　スキーマの設計

用例検索ツールの検索対象である文章データベースを格納するために設計したスキーマ（図３）は以下の特徴をもつ。

An Experimental Text Retrieval System on Kappa
Toshiyuki Okunishi, Hideo Miyoshi, Hiromi Abe, Yasuji Obuchi
Sharp Corporation

・テーブルを出典毎に分けることで、出典を制限した検索が可能になる。
・１文単位の追加、削除、修正が可能なように、１レコードを文番号と文から構成する。
・１文内の形態素数は可変であるため、形態素は繰返しとして定義する(list)。
・品詞、出現形、代表形は全ての形態素に存在することを指定する(not_nil)。
・複数個の値がとれる意味は繰返しとして定義する(list)。
・検索の高速化のために全ての文法素性に対しインデックスを設ける(index)。

```
_出典名(primitive, % 実際の出典名が入る
  文番号(type(integer),access(index)),
  文(list,
    品詞(type(string),access(index),
         value(not_nil)),
    出現形(type(string),access(index),
           value(not_nil)),
    代表形(type(string),access(index),
           value(not_nil)),
    種類(type(string),access(index)),
    活用型(type(string),access(index)),
    活用行(type(string),access(index)),
    活用形(type(string),access(index))
    意味(list,type(string),access(index)))).
```

図３：スキーマ

４．２　検索機能の実現

インデックス属性をもつ各文法素性をキーにしたインデックス検索により、指定キーワード中の全ての形態素を含む文は抽出できる。しかし、文を形態素の繰返し（すなわち集合）として定義しているだけなので、キーワード中の形態素の出現順序に関する条件まではチェックできない。そこで、次の２つの処理をするESPプログラムを補うことでLTB-KWICと同等の検索機能を実現した。

・形態素の出現順序がキーワードと異なる用例の排除
・１文中の全ての用例の抽出

５　評価

図３で示したスキーマに基づき1737文を格納するために必要な二次記憶容量は、18.3MB(うちインデックスは15.7MB)であり、LTB-KWIC版の4倍である。次に、検索時間の比較結果を表１に示す。LTB-KWICに比べ1.6～5.5倍かかっていることがわかる。

表１：検索時間の評価

	抽出用例数	LTB-KWIC	Kappa版
の+で+は+なく	7	3738	7385
仮定形	79	5081	8483
名+を+名+を	1937	37415	151969
名/人間+を~名/人間+を	5	13029	72767
名/自然物+が/格助	30	9058	40298

(単位:msec)

６　おわりに

Kappa版KWICでは１文単位の保守が可能なので、ユーザは大きな負担なくデータベースを変更できるようになり、データベースの保守が容易になった。しかし、５節で示したように、Kappaへ移植することで逆に検索時間が遅くなった。これはKappaが汎用のデータベースを対象としているのに対し、LTB-KWICが文章データベースに特化しているためである。

今後本システムを言語ツールとして実用的なものにしていくためには、効率も含めた検索機能の充実だけでなく、十分に有効な検索結果を抽出できるだけの大容量の文章データベースを装備していく必要がある。本稿で報告したKappa-IIへの移植実験によりKappa-P上での並列検索への見通しがついたことから、データベースが大容量になった場合にも十分に対処できる。

また文章データベースの大容量化に関して言えば、これまでのような人手による作成では開発量に限度があるため、文章を形態素切りし文法素性を付与するプログラムを現在ICOTで開発中である。

なお、本研究は第五世代コンピュータプロジェクトの一環としてICOTから委託を受けて行っているものである。

[謝辞]

日頃ご指導頂いたICOT第６研究室の田中室長、並びにKappaに関して有益なご意見を頂いたKappa開発グループの皆様に感謝致します。また、本ツールの開発・評価にご協力頂きましたｼｬｰﾌﾟﾋﾞｼﾞﾈｽｺﾝﾋﾟｭｰﾀｿﾌﾄｳｴｱ(株)の真田氏に感謝致します。

[参考文献]

[1]三吉,小渕,濱田,秋山,"構造化キーワードを用いた用例検索システムの試作", 情報処理学会 NL研 89-6, 1989.
[2]ICOT, 汎用日本語処理系LTB(第2版), 1990.
[3]ICOT, Kappa 1.1版 説明書, 1990.