3

# 結構やるな，KNP

黒橋 禎夫
京都大学大学院 情報学研究科知能情報学専攻
kuro@i.kyoto-u.ac.jp

## 構文解析の難しさはどこにあるのか

言葉は，表面的には音素や文字の1次元の並びであるが，内部には構造を持っており，それは次のような係り受け関係で表すことができる．



このような言語の構造を明らかにする計算機処理を構文解析と呼ぶ．構文解析は計算機による言語理解の第一歩であり，ある種のアプリケーションでは必須の処理である．たとえば，機械翻訳では，言語間の構造の違いを取り扱わなければならないので，構文解析が必要となるのはいうまでもない．

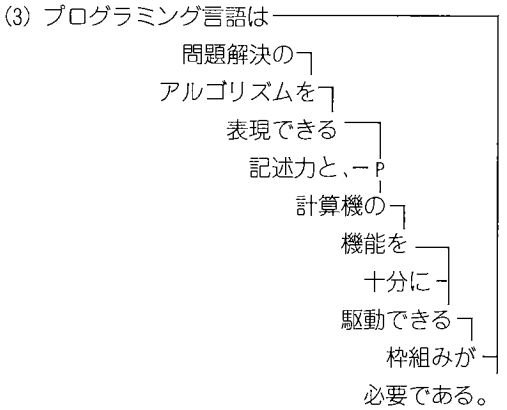
このようなことから，構文解析はこれまで自然言語処理研究の中心的な課題であったが，それは「一見簡単そうだが，実は非常に難しい」というのが研究者の認識であった．「一見簡単」というのは，人間がそれを苦もなく行っていることからくる素朴な直感である．しかし，上の2つの文の構造の違いを認識しようとすれば，意味，常識などの問題を取り扱うことが必要となり，それは「非常に難しい」ということになる．

たしかに，一昔前の構文解析の結果は「ひどい」ものであった．しかし，そのひどさの原因は本当に意味や常識の問題にあるのか？　この疑問が日本語構文解析システム，KNPの研究・開発の出発点であった．

文は何らかの意味・情報を伝えるものであり，先に意味を知っていなければ文構造が分からないというのはおかしな話である．文構造には優先される解釈というものがあり，そのような解釈に支えられて意味が運ばれる．

上のような係り受けの曖昧性（「クロール／望遠鏡で」が「泳ぐ」に係るか「見た」に係るかの曖昧性）では，近くの係り先（「泳ぐ」）が優先される解釈である．たとえば，「KNPで形態素解析した文を構文解析した」という文は，KNPをよく知る人でないかぎり「KNPで形態素解析した」と解釈するのが普通であろう．このような例は意味・常識の重要性をアピールする面白い例であるが，それが構文解析の難しさの第一の要因ではないはずである．

そうすると，いったい従来の構文解析結果がひどいのはどうしてなのか，ということをよく考えてみる必要がある．以下にいわゆるひどい解析例を示す（┐は通常の係り受け，┤P├は並列の関係を示す）．




（プログラミング言語は問題解決のアルゴリズムを表現できる記述力と、計算機の機能を十分に駆動できる枠組みが必要である。）

(4) 認可が
下り——P
次第、—P
出資を
予定している
メーカで
検討委員会を
作る。

(認可が下り次第、出資を予定しているメーカで検討委員会を作る。)

文(3)は「記述力」と「計算機」が並列していると解釈されている点が問題である．正しい構造は「問題解決のアルゴリズムを表現できる記述力」と「計算機の機能を十分に駆動できる枠組み」の並列であり，人間の場合には(計算機関係の専門家でなくても)並列のバランスなどを考慮してすんなり理解できるはずである．文(4)は「下り次第」という表現を正しく扱っておらず，「次第」を普通の名詞と扱っている点が問題である．そのため，「次第」と「出資」，「許可が下り」と「次第，出資を予定している」が並列であるという解析結果になってしまっている．

我々は，この2つの問題，すなわち並列構造の解析と，(単純な文法，少数の規則では扱えないという意味において)例外的な表現の取り扱いが構文解析においてまず解決しなければならない問題であると考えた．そこで，1991年から1993年にかけて，並列構造解析のアルゴリズムを開発し，これを基にして係り受け解析を行うKNPの最初のバージョンを作成した[1]．そして，1995年からは京都大学コーパスプロジェクトというプロジェクトを行い，その中で種々の例外的表現の働きを正しく取り扱うという文法の整備，システムの整備を重ねてきた[2]．

以下では，まずKNPの全体の枠組みを示し，次に上で述べた2つの問題について説明する．また，意味・常識の問題に対する我々の最近の取り組みも最後に簡単に紹介することにする．

## 解析の流れ

図-1にJUMAN，KNPによる解析の流れを示す．JUMANは，同じく京都大学で開発してきた形態素解析システムで，EDR辞書[3]を利用し，単語間の接続可能性に基づき入力文を単語ごとに区切り，各語の品詞，活用形などを同定する．

KNPでは，係り受けの処理単位を「1語以上の自立語(名詞，動詞など)＋0語以上の付属語(助詞，接尾辞など)」からなる文節という単位としている．そこで，まず，JUMANの出力結果である単語列を文節列にまとめる．後で説明する例外的な文節パターンをまとめることもここで行う．

KNPの処理の中心は，文節のさまざまな働きを示す属性を各文節に付与し，それに基づいて並列構造解析，係り受け解析を行う部分である．属性としては『体言』『用言』『ガ格』『ヲ格』『体言並列のキー』『用言並列のキー』などがある☆1．

文節に属性を付与するルールでは，その文節と，その前後の文節並びに対して条件を指定することができる．たとえば，

> 直後が「昔」「前」「先」のいずれかの名詞で始まる文節であり，『時間』属性を持つ「〜も」という文節であれば，『係:隣接』という属性を与える

というルールは次のように記述する．

```
((?*                          ;; 前の文節列に対する条件
 (< (?* [助詞***も]) ((時間)) >)
                              ;; 自分自身に対する条件
 (< ([名詞***(昔 前 先)]?*) >?*)
                              ;; 後ろの文節列に対する条件
    係:隣接)                  ;; 付与する属性
```

?*は，コンテキストによって，任意の文節列，または任意の単語列を示す．<>が文節の指定で，その中では文節を構成する単語列と文節の属性を指定する．[ ]が単語の指定で，品詞，品詞細分類，活用型，活用形，単語そのものの5つ組を指定する．単語指定中の*はその項が何でもよいことを示している．

並列構造解析，係り受け解析は，このようにして各文節に与えられた属性に従って行う．たとえば，(簡単化していうと)並列構造解析は「『体言並列のキー』の文節があれば，その前後の文節列から，末尾が『体言』文節で，最も類似する文節列を求め，並列構造の範囲とする」．また，係り受け解析は「『ガ格』文節は『用言』文節に係る」などの規則によって行われる．

☆1「本と鉛筆を買った」の「本と」のように，並列構造の存在を示す表現を並列構造のキーと呼ぶことにする．

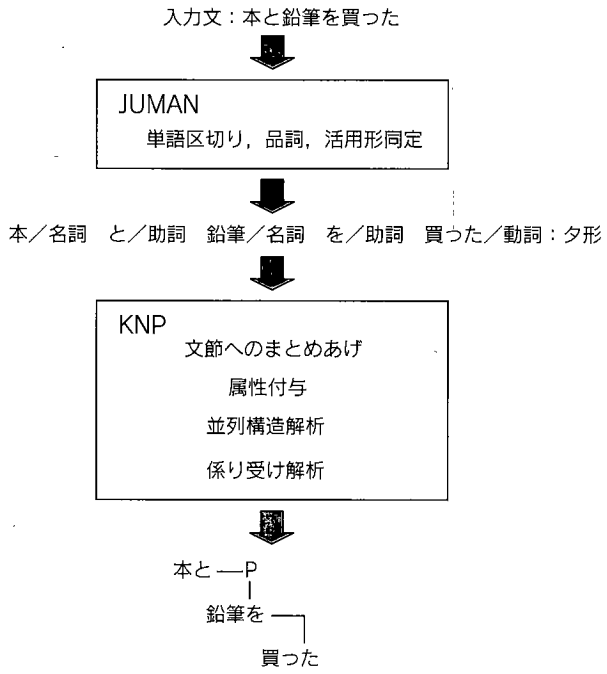



図-1　JUMAN，KNPによる日本語文の解析

## 類似性に基づく並列構造の扱い

「本と鉛筆を買った」「チーズを食べ，ビールを飲んだ」のような並列構造は，実際の文で非常によく現れる．たとえば，新聞記事では約40％の文が並列構造を含んでいる．並列構造の取り扱いの難しさは次のような点にある．

(1) 並列構造のキーの前後において，どの範囲が並列構造であるかという曖昧性がある．基本的な文法が与えてくれる制限は，キーが体言であれば並列構造の末尾も体言，キーが用言であれば並列構造の末尾も用言，ということだけであり，並列構造の範囲には多くの曖昧性がある．文 (3) の誤った解析例もこの基本的な制限は満たしている．

(2) ある表現が，並列構造のキーである場合とそうでない場合がある．たとえば，「サ変名詞＋読点」という表現には次のような曖昧性がある．

- 体言並列：「～が製造，販売について～」
- 用言並列：「～が製造，販売した～」
  「～が製造，～が販売した～」
  （「製造」の後ろに「し」（「する」の連用形）が省略されている）
- 従属節：「～が製造，……となるもようだ」

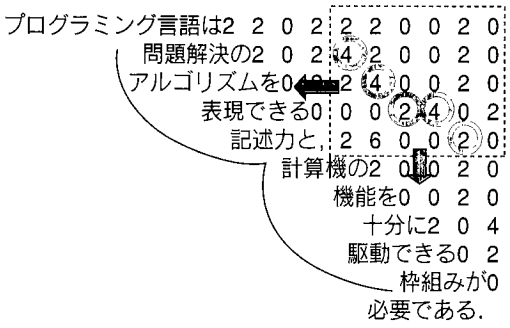



図-2　類似性に基づく並列構造の解析

(3) キーの種類が多い．「～と～」「～し～」のような典型的なものだけでなく，「～だけでなく～」「～であれ～であれ」等多数の表現がある．

KNPでは，このうち (1) と (2) の問題を，広い範囲の類似性を調べるという方法で解決している．すなわち，キーの前後で最も類似する文節列が並列の範囲であり，また，キーが曖昧な場合は，類似性の高い文節列があれば並列構造であると判断するのである．

文節列の類似性は図-2のような形で計算する．この図では，対角線上に文節が並んでおり，また，たとえば「問題解決の」という文節から右方向へ，「計算機の」という文節から上方向へ進んで交わった場所にある数字が，この2つの文節の類似度を示している．文節間の類似度は，自立語の一致，自立語の類似度（シソーラスにおける語の近さで近似），付属語の一致などから計算する．

並列のキーがあれば（図の例の場合「記述力と」），その右上部分（図の点線部分）における要素の並び（パスと呼ぶ）の中で，スコアが最大のパスをDP（動的計画法）によって求める．パスのスコアは，パス中の各要素の値の総和を，パスの長さで正規化したものとする．各要素の値はその左と下の文節間の類似度であるから，パスのスコアはその左と下の文節列間の類似度を示すものであり，このような方法によってキーの前後から最も類似度の高い文節列（＝並列構造の推定範囲）を求めることができる．たとえば図-2の例の場合には，この方法によって「問題解決のアルゴリズムを表現できる記述力と，計算機の機能を十分に駆動できる枠組みが」の部分が並列構造の範囲であると求まる．

表現が並列構造のキーであるかどうか曖昧である場合には，表現ごとに閾値を設け，最大スコアがこ

| キー*1 | 並列構造と認める条件*2 | 例 |
|---|---|---|
| {〜に} しろ | {〜に} しろ | 事件にしろ事故にしろ，〜 |
| {〜も} [用言条件形] | {〜も} 〜 | 個別経営もあれば協業もある． |
| [用言基本型]， | 〜と|〜など and 類似度4.0以上 | 〜では津波を警戒する，〜は開発を規制するなど〜 |
| {[連用の表現]} [サ変名詞]， | 直後の「[サ変名詞] ＋する」という文節 or 類似度3.5以上 | 〜候補を推薦，支持しており〜十八日にも新党準備会を結成，二月中にも旗揚げする方針 |
| 〜 [格助詞]， | 同じ格助詞 and 類似度4.0以上 | ショベルカーの本体は日本で，先端部分はメキシコで〜 |
| {〜は} もちろん， | 〜も｜でも｜まで | 兵士の家族はもちろん，一般の国民も〜 |
| 〜だけ｜のみ｜ばかりで（は）なく | | 〜を大幅に上回っただけではなく，…記録をも更新した． |

*1 {} は直前の文節．
*2 並列構造末尾の表現に対する条件と，類似度に対する閾値．

**表-1** 並列構造のキーとその扱い

の閾値を超える場合には並列構造であり，超えない場合にはそうではないと判断する．

キーの種類の多さの問題については，基本的に丁寧に文法を整備するしかない．この問題は次の章で述べる方法で対処した．表-1にキーとその扱いの一例を示す．

## 例外的表現の扱い

日本語には（単純な文法，少数の規則では扱えないという意味において）例外的表現が多く，はじめに示した文(4)の解析結果のように，それらをきちんと扱わなければ文の解析が大きく誤るということになる．

しかし，このような問題は，個々の表現の問題であるので，その働きをきっちりと指定すればよく，副作用等に気を使う必要はあまりない．問題は，そのような表現をいかに収集するかという点にある．

この問題は，京都大学コーパスプロジェクトにおいて扱った．このプロジェクトは，KNPで大量の文を解析し，その結果を人手で修正し，正しい構文情報を付与したコーパスを作成・公開するというものである☆2．この，人手による構文解析結果の修正の際に，例外的な表現を発見し，その働きを随時文法に取り込んでいくということを行った．新聞記事を対象に4万文のコーパスを作るという過程で，数十タイプの例外的表現を発見した．それらの一例を以下に示す．

**文節の多様性**：「自立語1語（名詞は2語以上の連続を許す）＋付属語0語以上」という基本パターンではないが，文節としてまとめてしまった方が働きを記述する上で都合がよいもの．

「書き損じる」「読み飛ばす」
「書かざるをえない」「書いたに違いない」「書くとはいえ」
「書くのかどうかさえ」「書かずじまい」「書こうものなら」
「書いた途端｜矢先」「書いて以降｜以来｜以後」
「書き次第」

**係り受けの多様性**：係り先が（単純な意味での）用言でないもの．

- 判定詞を補うもの
  「(〜が) 学生.」(文末の体言止)
  「(〜が) 学生かどうか」「(〜が) 学生とはいえ」
  「(〜が) 学生の場合」
- サ変名詞を動詞と解釈するもの
  「(〜を) 発表.」(文末のサ変名詞の体言止)
  「(48キロ級で) 連勝中の」「(昨日) 公開の」
- 相対名詞，時間表現，数量表現など
  「(〜より) 右」「(〜から) 上」
  「(〜から) 3日間」「(年に) 3度」「(30年も) 前に」
- その他

☆2 現在このコーパスは，統計的自然言語処理の訓練・評価データとして多くの研究機関で利用されている．

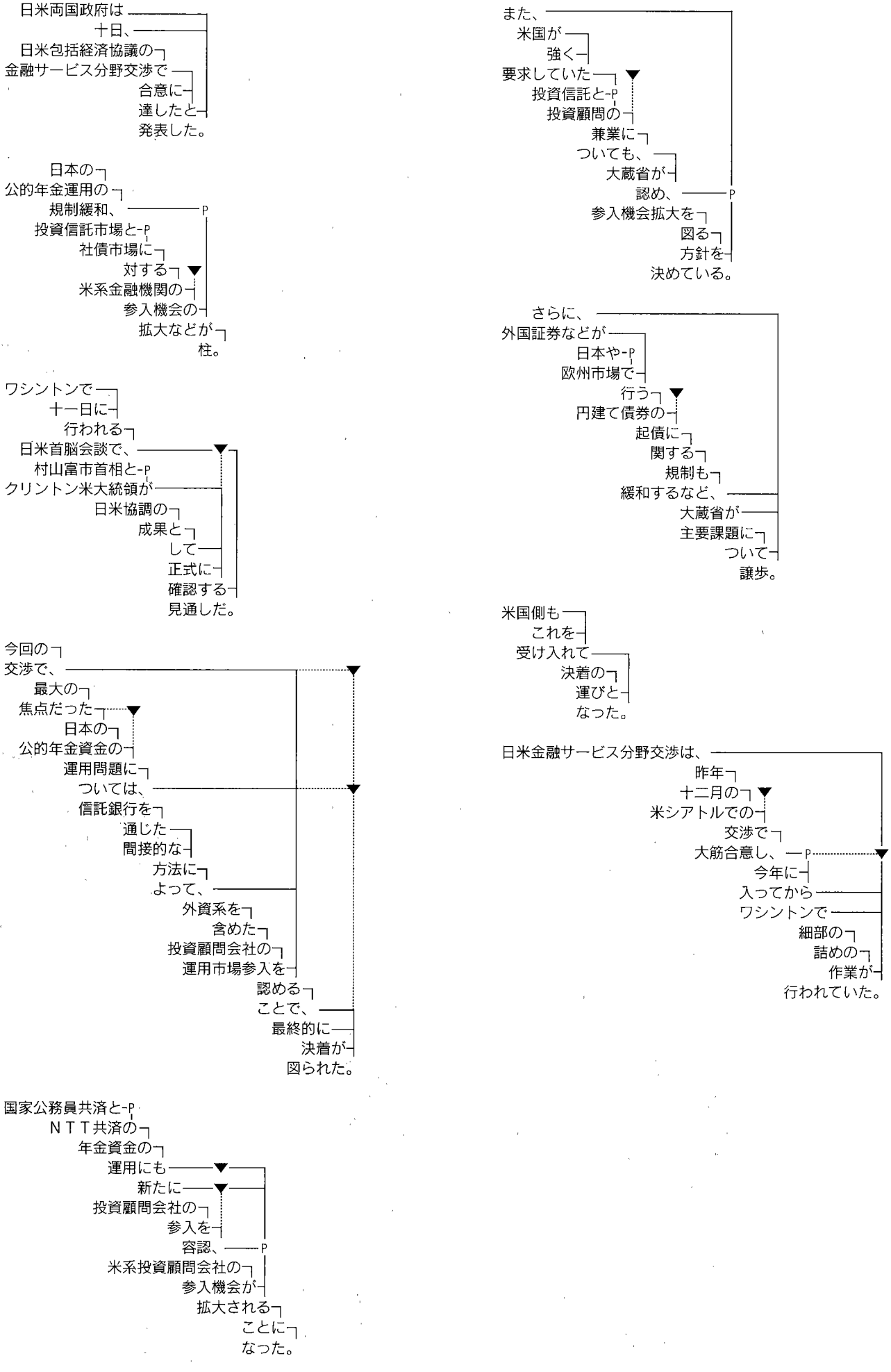



図-3　KNPによる新聞記事解析例（▼は，自動解析が誤りの場合の，正しい係り先を示す）

「(～を) 中心に」「(～を) 基に」(「して」の省略と解釈)

## 解析の精度

図-3に新聞記事の解析例を示す．意味・常識を知らなくても，構文解析は「結構うまくいく」ものである．

構文解析の精度は文章のタイプに大きく左右される．京都大学コーパスプロジェクトにおいて正しい構文情報を付与した新聞記事テキスト (4万文) と，JUMAN，KNPの自動解析結果を比較すると，精度は以下のようになる．

- 形態素解析 (単語区切りと品詞同定)：99％
- 文節区切り：99％
- 係り受け解析：90％ (一般記事) ～92％ (社説)
  (文末および文末から2番目の文節を除いて，文節の正しい係り先が得られた割合)

これらの値は「新聞記事的文章」に対するJUMAN，KNPの解析精度の目安となる．一般記事と社説を比べると，一般記事の方がコンパクトな (詰め込んだような) 表現が多く，解析精度が若干低い．科学技術論文，特に抄録などは，この一般記事程度の解析精度になると予想される．

Eメール，ネットニュースなどの文章は，口語的で，解析も難しくなる．しかし，新聞記事の中にも引用文として口語的表現がかなり含まれており，KNPはそれらについてもある程度対処している．たとえば「それって無理」という文の場合，「って」が助詞であること，また，文末の省略表現の扱いによって「それって」が「無理」に係ることが正しく解析される．

一方，音声認識結果，あるいは講演などの書き起し文など，「話し言葉」をそのままテキストにしたものの場合には，解析は相当難しい．1つは言い淀み，言い直しなどの問題であるが，それよりも，そもそも文として成り立たないような発話が問題になる．それらは，次に述べる「語と語の関係」に基づく解析が重要になり，「書き言葉」でいうところの構文解析よりも文章解析に近いレベルの処理が必要となるはずである．

## 今後の方向

我々が現在取り組んでいる問題は，語と語がどのような関係を持ちやすいかということを学習し，その情報に基づいて解析のレベルを上げることである．語と語の関係とは，たとえば「クロールで泳ぐ」「望遠鏡で見る」などの関係である．このような関係は，現在のKNPで大量の文を解析し，その結果から信頼できる部分を集めることによって自動的に得ることができる[4]．この情報は，係り受けの曖昧性を解消するだけでなく，格解析，省略・照応解析などで利用することができ，文の解析だけでなく，文章の解析というレベルにつながっていく．

はじめにも述べたように，構文解析は計算機による言語理解の第一歩であり，機械翻訳などのアプリケーションでは必須のものである．一方，情報検索などの，情報へのアクセスを高度化することに現在の構文解析がすぐに役立つかというと，それはまだ難しいかもしれない．しかし，KNPを発展させ，格解析，省略・照応解析などが「結構うまくいく」レベルになれば，役立つことは間違いないだろう．そのようなシステムの実現はそう遠くないと考えている．

JUMAN，KNPは研究利用の場合には無償で利用可能であり，http://www-nagao.kuee.kyoto-u.ac.jp/nl-resource/から入手できる．動作環境はUNIX系OS，Windowsである．また，ダウンロードをしなくても，このページにある「実践KNP」で任意の文を打ち込めば，その解析結果を見ることができる．

**参考文献**
1) 黒橋禎夫, 長尾　真: 並列構造の検出に基づく長い日本語文の構文解析, 自然言語処理, Vol.1, No.1 (1994).
2) 黒橋禎夫：コーパスが先か，パーサが先か, 情報処理, Vol.41, No.7 (July 2000).
3) 日本電子化辞書研究所: EDR電子化辞書仕様説明書 (1993).
4) 河原大輔, 鍛治伸裕, 黒橋禎夫: 大規模コーパスからの格フレーム辞書構築とそれを用いた格解析, 言語処理学会 第5回年次大会, pp.24-27 (2000).

(平成12年9月27日受付)