TR-IT-0142

音声言語処理のための構文解析ツールキット ユーザーズマニュアル

A Parsing Toolkit for Spoken Language Processing User's Manual

田代 敏久
Toshihisa Tashiro

1995.12

概 要

本報告書では、音声言語処理の効率的な研究を可能にする構文解析ツールキットの使用法、プログラムの改造 (改善) のためのヒント、今後の研究課題等について述べる。

ATR 音声翻訳通信研究所
ATR Interpreting Telecomunications Research Laboratories

もくじ

第 1 章

構文解析ツールキットマニュアル

1.1 はじめに

- 本マニュアルは、 UNIX と C(及び基本的な LISP) の知識を持つ人を対象に書かれています。

- 構文解析ツールキットは、特定の文法理論や文法体系に依存していません。したがって、本マニュアルには品詞名や句構造規則等の説明は含まれていません。

- 構文解析ツールキットには、解析実験を行なうためのデータベースや文法も含まれています。このデータや文法は、基本的に ATR で作成している音声言語データベースに基づくものですが、完全に同一なものではありません。

- 構文解析ツールキットの各プログラム中には、本来可変にすべきパラメータが定数として埋め込まれてしまっています。これは、単に作者の怠慢によるものですが、環境変数や rc ファイルを乱用して混乱するより、定数を変更してコンパイルしなおす方がましだということも事実でしょう。

- ある意味において、構文解析ツールキット (このマニュアルも含めて) は“閉じて”います。つまり、プログラムもデータも文法もドキュメントも[1]すべて一つのパッケージとしてまとめられており、実験や改造のために必要な外部資源は、エディタ、シェル、C コンパイラ (GCC もしくは ANSI C 準拠の C コンパイラが必要です) ぐらいなものです。この構文解析ツールキットを利用・理解するために必要な資源は、本ドキュメント、各ディレクトリにおかれた README ファイル、ソースファイル、UNIX 及び C に関する知識、そしてあなたの想像力だけです。

[1]混乱をさけるために、lisp の実行イメージさえも含まれています。

- 別な意味では、構文解析ツールキットは完全に “open” だとも考えられます。各プログラムは、いずれも単純なアルゴリズムを利用しているので、少し努力すればプログラムを読解することも可能でしょう。また、単純な入出力インターフェースは、すべて単純なテキストファイルを利用していますので、 sh や csh のプログラムができる人なら、各ツールの結合は容易なはずです。

1.2 構文解析ツールキットのインストール

/dept4/work2 にある ptk.tar を適当なディスクに展開してください。

次ページ移行、$PTK を 展開先のディレクトリに置き換えて読んでください。

構文解析ツールキットには、hp-ux で動作する実行ファイルが含まれています。他の 機種や OS で動作させる場合には、各ディレクトリで make clean した後、make しなおしてください。ただし、どのプログラムもかなりのメモリを必要とします。

1.3 形態素解析プログラム

1.3.1 形態素解析とは

形態素解析とは、入力文字列を単語単位に分割し、各単語に品詞を付与する処理を意味します。通常、文字列の単語分割や品詞付与には曖昧性が生じますので、なんらかの方法を用いて曖昧性を解消する必要があります。このプログラムは、品詞のバイグラムとビームサーチを用いて曖昧性の解消を試みます。

1.3.2 実行方法

$PTK/morph にカレントディレクトリを移動し、README ファイルを読んで下さい。

1.3.3 必要な言語知識

この形態素プログラムを動作させるには、単語辞書、品詞のバイグラム、品詞別単語出現確率の 3 つの言語知識が必要です。

単語辞書

単語辞書は

```
辞書見出し 品詞ラベル 読み
:
:
```

という形式のテキストファイルとして用意します。

なお辞書見出しには、2 バイト文字しか利用できません。

品詞ラベルには、スペース文字やタブ文字以外の任意の印字可能な利用可能です。なお、品詞ラベル中に "-"(ハイフン) が含まれていた場合、ハイフンより前の文字列のみを出力することが可能です (解析の動作自体には影響を与えません)。

読みはカナ表記とするのが一般的だと思われますが、品詞ラベルにと同様、ASCII 文字を利用しても構いません。

単語辞書の具体例は、$PTK/morph に置かれている LEX というファイルを参照してください。

品詞のバイグラム

品詞のバイグラムは、

```
前の品詞ラベル 着目する品詞ラベル 確率の対数値
:
:
```

という形式のテキストファイルとして用意します。

品詞のバイグラムの具体例は、$PTK/morph に置かれている POS というファイルを参照してください。

品詞別単語出現確率

品詞別単語出現確率は、

```
辞書見出し 品詞ラベル 確率の対数値
:
:
```

という形式のテキストファイルとして用意します。

品詞別単語出現確率の具体例は、$PTK/morph に置かれている WOP というファイルを参照してください。

1.3.4 各種オプションについて

この形態素プログラムには、以下のようなコマンドラインオプションがあります。オプションの指定順序は任意です。

- -D
  デバッグモードで動作します。入力文字列中の単語候補のラティスを表示します。
- -B number
  グローバルビーム幅を指定します。デフォルト値は 100 です。
- -b number
  ローカルビーム幅を指定します。デフォルト値は 10 です。
- -n number
  出力する候補数を指定します。デフォルト値は 1 です。

- -f number

  出力形式を指定します。出力形式の例を以下に示します。

```
[number = 1](デフォルト)
((出力 サ変名詞)(形式 普通名詞)(を 格助詞)(指定 サ変名詞)
(し 補助動詞)(ま 助動詞)(す 語尾)(。 記号))


[number = 2]
((出力 サ変名詞 -<>-<>)(形式 普通名詞 -<>-<>)(を 格助詞 -<>-<>)
(指定 サ変名詞 -<>-<>)(し 補助動詞 -<サ変>-<連用>)
(ま 助動詞 -<特殊サ>-<語幹>)(す 語尾 -<特殊サ>-<終止>)
(。 記号 -<>-<>)(score = -26.731845))


[number = 3]
((出力 シュツリョク サ変名詞 -<>-<>)(形式 ケイシキ 普通名詞 -<>-<>)
(を ヲ 格助詞 -<>-<>)(指定 シテイ サ変名詞 -<>-<>)
(し シ 補助動詞 -<サ変>-<連用>)(ま マ 助動詞 -<特殊サ>-<語幹>)
(す ス 語尾 -<特殊サ>-<終止>)(。 。 記号 -<>-<>)(score = -26.731845))


[number = 4]
シュツリョクケイシキヲシテイシマス。

[その他を指定した場合]
1| 見出し | 出力 | 形式 | を | 指定 | し | ま | す |。 |
1| 品詞 | サ変名詞 -<>-<>| 普通名詞 -<>-<>| 格助詞 -<>-<>| サ変名詞 -<>-<>
| 補助動詞 -<サ変>-<連用>| 助動詞 -<特殊サ>-<語幹>
| 語尾 -<特殊サ>-<終止>| 記号 -<>-<>|
```

1.3.5 未知語処理について

この形態素プログラムは、非常に単純な未知語処理を行なっています。
未知語処理の概要は以下のとおりです。

- 入力文字列中の任意の１文字は単語を形成することができる。
- 入力文字列中の連続したカタカナ及びアルファベット文字列は単語を形成することができる。
- 未知語の品詞の推定は行なわない。

品詞別単語出力確率が低い場合、辞書登録されている語でも未知語として解釈されることもありうることに注意してください。

1.3.6 プログラム改造のポイント

辞書やバイグラムファイル名を変更したい場合

$PTK/morph/define.h 中の定数を適宜変更してください。

出力形式を変更したい場合

$PTK/morph/morph.c 中の print_candidate という関数を変更してください。

未知語処理メカニズムを変更したい場合

$PTK/morph/morph.c 中の make_src_lattice という関数を変更してください。

1.3.7 研究上の課題

この形態素解析プログラムをより高度にするための課題としては以下のようなものが挙げられます。

- 入力の前処理：
  この形態素解析プログラムは、ビームサーチを利用しているために、入力文が長くなるほど解析精度が悪くなります。入力文を的確に分割することにより、精度の向上が見られるでしょう。

- インクリメンタルな解析：
  この形態素解析プログラムは、アルゴリズムを単純にするために、入力文字列のすべてを辞書引きしてから探索を始めています。本当は、探索時に動的に辞書引きを行なった方が効率がよくなるででしょう (この改造はそれほど難しくないずです)。また、ラティスを動的に拡張するメカニズムを設けることにより、入力を本当にインクリメンタルに解析することも可能でしょう。1 pass のビームサーチの利点は、インクリメンタルな解析を可能にすることなのです。

- 2(multi) pass サーチの実現：
  逆に、(multi) pass サーチを実現すれば、より高い精度が実現できる可能性があります。また、バイグラムの計算のパックが可能になりますから、全体の計算時間も短くなる可能性があります。

- 未知語処理の改良：
  文字の統計値等を利用すれば、より精度の高い未知語処理が可能になる可能性があります。

1.3.8 LISP 版形態素プログラムについて

C版の形態素プログラムと基本的に同等のアルゴリズムで動作する LISP 版形態素プログラムが、$PTK/LISP/morph に収められてます。

LISP 版では、曖昧性の解消を “単語数最小優先” という単純なヒューリスティックで行なっています。バイグラム等の統計データが入手できない場合は、LISP 版を利用すれば素早く形態素解析システムを作ることができるでしょう。

1.4 形態素調整

形態素調整とは、異なる形態素体系の差異を自動的に解消する処理を意味します。この処理には、

- 形態素の差異をコーパスからルールとして抽出する機能
- 抽出したルールを実際に適用し、形態素体系の差異を自動的に解消する機能

の二つが必要です。

1.4.1 ルールの抽出

実行方法

$PTK/madjust/Extracting_Rules の README ファイルを読んでください。出力されたルールに付与されている数値は、規則の左辺別のルール出現確率ですが、今のところ無視してもらって結構です (ただし消去しないでください)。

動作原理

ルールの抽出は、単純なパターンマッチを利用しています。後に説明する形態素解析評価ツールと同じ動作原理です。

1.4.2 形態素体系の差異の解消

実行方法

$PTK/madjust/Rewriting の README ファイルを読んでください。

必要な言語知識

形態素調整ルールと品詞別単語出力確率及び品詞バイグラムが必要です。形態素調整ルールは、以下のような S 式で用意します。

```
((旧辞書見出し 旧品詞ラベル) <--> (新辞書見出し 新品詞ラベル))
                    :
                    :
```

品詞別単語出力確率及び品詞バイグラムは、通常の形態素解析プログラムと同様です。

形態素調整ルールの具体例は、$PTK/madjust/Rewriting に置かれている RULEFILE というファイルを参照してください。

動作原理

動作原理は通常の形態素解析とほとんど同じです。ただし、ローカルビームサーチは行なっていません。

1.4.3 各種オプションについて

この形態素プログラムには、以下のようなコマンドラインオプションがあります。オプションの指定順序は任意です。

- -b number
  (グローバル) ビーム幅を指定します。デフォルト値は 100 です。
- -n number
  出力する候補数を指定します。デフォルト値は 1 です。
- -W number
  品詞別単語出力確率の重みを指定します。 他の重みとの相対値がスコア計算時に利用されます。 デフォルト値は 1 です。
- -P number
  品詞バイグラムの重みを指定します。 他の重みとの相対値がスコア計算時に利用されます。 デフォルト値は 1 です。
- -R number
  ルールのスコアの重みを指定します。 他の重みとの相対値がスコア計算時に利用されます。 デフォルト値は 0 (ルールのスコア無視) です。
- -r filename
  ルールファイル名を指定します。デフォルト値は “RULEFILE” です。
- -w filename
  品詞別単語出力確率のファイル名を指定します。デフォルト値は “WOP” です。
- -p filename
  品詞バイグラムのファイル名を指定します。デフォルト値は “POS” です。

1.4.4 研究上の課題

この形態素調整プログラムをより高度にするための課題としては、ルールのスコアの有効利用が挙げられます。既にメカニズム上はルールスコアを扱えますので、スコアの計算方法がポイントとなります。

1.5 句構造解析

1.5.1 句構造解析とは

構文解析とは、形態素列を句構造文法を利用して木構造を作成する処理を意味します。この場合も曖昧性が生じますし、また一つの木構造にまとまらない場合もあります。

このプログラムでは、曖昧性の解消は特に行なわず、全解探索を行ないます。また、メモリが不足したり、文法上一つの木にまとまらない場合には、チャート中の部分木を探索することが可能です。

1.5.2 実行方法

$PTK/cfg の README ファイルを参照してください。

1.5.3 プログラム改造上のポイント

- 出力形式を変更したい場合：

- 全解探索でなく、n-best 探索を行ないたい場合：
  chart_suspend_p という関数を変更してください。

- チャート中の部分木検索方法を変更したい場合：

  部分木の探索方法 (ビームサーチ) は、形態素解析での探索方法とほとんど同一です。
  chart_search.c というファイルを変更してください。

1.5.4 研究上の課題

このパーザは、いわゆるアジェンダコントロールをまったく行なっていません。効率な n-best 探索を行なうためには、chart_get_pending_edge というマクロを関数を変更し、何らかの順位づけを行なう必要があります。有望なのは、CFG 規則の統計情報です。$PTK/cfg に RULE-BIGRAM という統計情報ファイルがあります。これは、句構造規則の適用順序のバイグラムを計算したものです。

1.6 その他の解析モジュール等

1.6.1 構文木調整

句構造の差異を調整するプログラムです。$PTK/badjust の btraining というプログラムが調整ルールを学習し、bparse で実際に解析を行ないます。README ファイルを参照してください。

1.6.2 依存構造解析

句構造解析結果を依存構造にするプログラムです。

C 版が $PTK/depend に、LISP 版が $PTK/LISP/dependency にあります。それぞれの README ファイルを参照してください。

1.6.3 格構造解析

依存構造を格構造にするプログラムです。LISP 版のみ $PTK/LISP/CASE-parser にあります。README ファイルを参照してください。

1.6.4 意味解析 (木構造書き換え)

任意の木構造を変換するプログラムです。LISP 版のみ $PTK/LISP/rewriting にあります。README ファイルを参照してください。

1.6.5 各種のユーティリティ

C 版 のユーティリティが $PTK/tools に、LISP 版のユーティリティは $PTK/LISP においてあります。

特に重要なツールは 形態素及び句構造の評価ツールです。

$PTK/tools の meval.[1-3] が形態素の評価ツール です。通常は meval.3 を用いてください。

句構造の評価は、まずパーザの出力を正規化し ($PTK/tools/normalize を用いてください)、その出力を beval で処理してください。

1.6.6 実験用データベース

$PTK/DATA に各種のデータベースを整理してあります。README ファイルを参照してください。

第 2 章

参考資料:NL 研資料より抜粋

2.1 はじめに

音声翻訳システムのような音声言語処理システムの構築のためには、頑健で解析精度が高く、処理効率も良い構文・意味解析機構の研究が必要である。また、単に解析能力の優劣だけではなく、他の処理モジュールとの協調的な動作が可能かどうかということも、重要な構文・意味解析機構の評価尺度である。しかし、従来の構文・意味解析の研究の多くは、以下の 3 つの型に分類することができ、それぞれに問題を抱えている。

- 文法理論重視型:
  構文・意味解析の研究に、文法理論や他の言語学上の研究成果を採り入れるのは当然である。しかし、少なくとも現状の文法理論は、現実世界の多種多様な言語現象を矛盾なく説明できるほど緻密ではない。よって過度に特定の文法理論に依存していては、頑健な解析機構の開発は不可能である。

- 計算メカニズム重視型:
  構文・意味解析の研究に、ソフトウェア科学上の成果を採り入れることも重要かつ必要である。しかし、構文・意味解析機構は必然的に大量のデータ処理を必要とするのに対し、ソフトウェア科学上の新しい計算メカニズムは、しばしば現実には不可能な資源 (メモリおよび計算量) を要求するために、実際に動作する構文・意味解析機構の研究には向かないことが多い。また、新しい計算メカニズムは、記号処理の世界に閉じていることが多く、音声言語システムの開発に必要となる記号処理と非記号的情報処理との協調作業を困難にしている。

- 開発重視型:
  上記の 2 つの型が理論重視なのに対し、もっぱら実践を旨とする研究も存在する。こ

の種の研究では、かなり解析精度が高く、処理効率も良い結果を得ていることが多い。しかし、この種の研究は、ある特定のシステムに依存した処理機構やデータを前提としていたり、ドメインに依存したヒューリスティックを無批判に利用していたりする場合もある。また言語学的基盤や計算メカニズムが明確でないため、実験結果を客観的に評価することが困難な場合が多い。

我々は従来、文法理論としてはJPSGのような制約に基づく句構造文法、計算メカニズムとして単一化演算を基礎に、音声言語解析の研究を進めてきた[4]。しかし、特定の文法理論や計算メカニズムに深く依存した研究は、前述のような問題を持ち、我々が目標とする“自発的に発声された話し言葉(spontaneous speech)の処理”を行なうためには不十分であることがわかってきた。そこで、我々は構文解析機構の研究・開発を、以下のような方針で行なうことにした。

- 制約に基づく句構造文法の枠組を守りながらも、他の文法理論や言語学上の知見、コーパスから自動的に学習された統計的・計量的な知識等を積極的に採り入れ、話し言葉に出現する幅広い言語現象を処理できる文法、言語モデルを開発する。
- 単一化演算をそのまま実装、利用するのは計算コスト、多様な言語的知識の柔軟な利用、等の点で問題があるので、より処理効率が良く、改良・改造や他のモジュールとのリンクが容易な計算機構を用意する。

本稿では、上記の方針に従って整備を進めている構文解析ツールキットの概要、および予備的な言語解析実験の結果を報告する。

2.2 構文解析ツールキットの概要

2.2.1 目標とする解析機構のイメージ

前述のように、単一化文法に基づく構文解析は、多様な言語的知識を柔軟に利用したり、他のモジュールとリンクしたりすることが困難である。これは、単一化文法に基づく構文解析機構は、基本的に図2.1で示すような硬直化した設計思想に基づいて作成されているためである。

このような設計思想で作成された構文解析機構には、以下のような問題がある。

- 文法・辞書には、統語的な知識や意味的・運用論的な知識を混在させて記述する必要があるので、大規模な語彙や言語現象をカバーすることが困難である。

入力：文字列

↓

[単一化パーザ]

↓

出力：意味表現

Figure 2.1: 単一化文法に基づく構文解析のイメージ

- 出力が固定されているため、出力構造と相性の悪い外部モジュールは、パーザの出力を利用できない。

そこで、我々は図2.2で示すような、1) 処理機構および知識がモジュール化されており、2) さまざまなレベルの出力を外部モジュールに提供可能な、構文・意味解析機構を目指す。

2.2.2 開発にあたっての留意点

我々は、以下のような事項に留意して構文・意味解析機構を開発している。

- 処理機構と知識を分離する
  単一化に基づく構文解析のように、計算機構および知識の記述形式が統一されている場合には、特に注意しなくても処理機構と知識は形式的に分離される。しかし、我々の枠組は、計算機構や知識の記述形式の多様性を認めるので、不注意により処理機構の中に知識が埋め込まれてしまう恐れがある。

- 制約と選好を区別する
  単一化に基づく構文解析では、すべての言語的知識は制約として扱われる。しかし、長尾ら [3] が述べているように、制約としての知識と選好としての知識を区別して考慮しなくては、頑健で高精度な解析機構は開発できない。

- データ (構造) の書き換え (変換) のための手続き・知識と、曖昧性解消のための手続き・知識を区別する
  単一化文法に基づく構文解析に限らず、従来の構文解析の研究の多くは、曖昧性解消

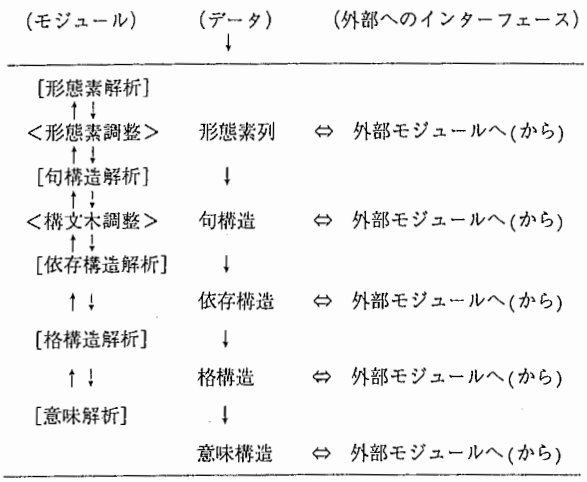

Figure 2.2: 望ましい構文解析機構のイメージ

の重要性を強調するあまり、解析の基本的な手続き・知識と曖昧性解消のための手続き・知識を区別することを怠ってきた。本当は、曖昧性解消の研究を効率的に進めるためにこそ、両者を区別することが重要なのである。我々が開発している構文解析ツールキットのすべてのモジュールは、以下の原則を遵守しているので、曖昧性解消の研究をより柔軟に行なうことができる。

- 各モジュールは極めて単純なデータの変換・書き換え機能 (=基本機能) のみを有する。
- 曖昧性の解消は、各モジュール固有のヒューリスティックや統計情報、他のモジュールの処理結果等を利用して、基本機能とは分離可能な計算機構により行なう。

2.2.3 各モジュールの概要

前述の方針や原則に基づき開発を進めている各モジュールの概要を説明する。なお図 2.3に、各モジュールの基本機能, 基本機能に必要な知識、曖昧性の解消に必要な知識についてまとめる。

形態素解析モジュール

形態素解析モジュールとは、入力文字列を単語単位に分割し、品詞・活用形等の形態素情報ラベルを付与するモジュールである。辞書を入れ換えることにより、任意の形態素情報体系のもとで動作する必要がある。また、形態素解析時に発生する曖昧性の解消には、様々な

知識が提案されているので、それらの知識を有効に利用できるメカニズムを用意する必要がある。図2.4に形態素解析モジュールの入出力例を示す。

形態素調整モジュール

形態素調整モジュールとは、ある形態素情報体系 (単語の分割および品詞情報) に基づく形態素列を、別な形態素情報体系に基づく形態素列に書き換えるモジュールである。このモジュールは本来、異なる形態素情報体系で作成された形態素情報コーパスを有効利用するために開発されたが [5]、形態素解析と句構造解析の間での形態素情報体系の調整にも利用できる。このモジュールも、形態素解析モジュールと同様、任意の形態素情報体系で動作し、曖昧性解消のための様々な知識を利用できなくてはならない。図2.5に形態素調整モジュールの入出力例を示す。

句構造解析モジュール

句構造解析モジュールとは、形態素列を入力にとり、句構造規則に従い、構成要素 (constituent) をノードとする木構造 (句構造) を作成するモジュールである。

我々は、句構造解析に文脈自由文法を用いている。自然言語の持つ文脈依存性は、句構造解析モジュールとは独立した計算機構および知識を利用して対応する。図2.6に句構造解析モジュールの入出力例を示す。

構文木調整モジュール

構文木調整モジュールとは、ある構文解析文法に基づく句構造 (構文木) を、別な句構造 (構文木) に書き換えるモジュールである。このモジュールは本来、異なる構文解析文法で作成された構文木コーパスを有効利用するために開発されたが [6]、句構造解析と依存構造解析間での構文解析文法の調整にも利用できる。図2.7に構文木調整モジュールの入出力例を示す。

依存構造解析モジュール

依存構造解析モジュールとは、句構造 (構文木) を入力とし、構成要素 (constituent) の統語的な主従関係を判定し、“語”[1]をノードとするラベル無し有向グラフ[2]に変換するモジュールである。

---

[1]通常は単語と考えてよい。動詞等の活用する語は、複数の形態素で一つの単語を構成すると考える。

[2]ほとんどの場合、木構造で十分表現できる。

一般的には、 このモジュールは、次に述べる格解析モジュールと一体をなすものとして考えられているが、我々はできる限りモジュールを細分化する方針なので、あえて 1 モジュールとして独立させている。図 2.8に依存構造解析モジュールの入出力例を示す。

格構造解析モジュール

格構造解析モジュールとは、依存構造 (ラベル無し有効グラフ) を入力とし、格解析辞書を利用して “語” の主従関係の役割を決定し、格構造 (ラベル付き有向グラフ) に変換するモジュールである。

なお、このモジュールはあくまでノード間の関係を決定するだけで、主従関係の逆転や、表層にない要素の生成等は行わない。格構造のノードは必ず表層の語と直接的な対応関係を保っている。図 2.9に格構造解析モジュールの入出力例を示す。

意味解析モジュール

意味解析モジュールとは、ラベル付き有向グラフを、グラフ書き換え規則を利用して変形するモジュールである。

このモジュールは、主従関係の逆転、ノードの挿入・削除等、任意の変形操作を行うことができる。結果として、意味解析結果と表層の語とは直接的な対応関係は失われてしまう。図 2.10に意味解析モジュールの入出力例を示す。

2.2.4 自由発話への対応

我々の解析対象は話し言葉なので、必ずしも文法的に適格な文が入力されるわけではない。 そこで、不適格な入力を処理するために、 Jensen[1] により提案された Fitted Parse の手法を用いることにした。Fitted Parse とは、

- 適格文を対象とする核文法 (core grammar) を用いてボトムアップに統語解析する。
- 統語解析に失敗した場合、保存されている途中結果 (部分木) を出力し、後続の処理に委ねる。

という手法である。

この手法を実現するために、我々の構文解析ツールキットは以下のような特徴を持っている。

- すべてのモジュールは、入力に対し、機械的に定義できないような制約 (“入力は文でなくてはいけない”、等の制約) を科さない。

- すべてのモジュールは、処理に失敗した場合でも、部分的な解を出力することができる。

- 文字列、形態素列、句構造等、各レベルにおいてデータの分割・併合ツールを用意し、処理単位を再構成できるようにする。

2.3 解析実験

前節で概説した構文解析ツールキットは、まだ開発途上にある。しかし、句構造解析モジュール等は既にほとんど完成しているので、極めて簡単な解析実験を行なってみた。

解析対象は、ＡＴＲで作成している音声言語データベース [8] に含まれている 2352 文 (64 会話) である。これらの文は既に形態素解析および句構造解析が済んでいるので、形態素解析結果を入力とし、句構造解析結果を正解ファイルとすることにより、句構造解析実験を容易に行なうことができる。なお、これらの文は必ずしも文法的に適格ではないので、図 2.11で示すような、部分木 (の集合) としてしか解釈できない文もかなりある。

使用した文法は、適格な日本語文を想定して作成された 229 規則からなる純粋な文脈自由文法 (核文法) である。前節で述べたように、我々の構文解析ツールキットは部分的な解を出力することができるので、適格文のみを想定した文法を利用しても、必ず何らかの解析結果を得ることができる。

純粋な文脈自由文法を用いた構文解析では、当然のことながら大量の曖昧性が生じる。我々は、曖昧性解消のための知識として文法規則の統計情報を用いることにし、解析対象とは別の 4237 文 (104 会話) から、1) 通常の確率文法と、2) 文脈依存の確率文法の一種である北 [2] により提案された言語モデルの 2 つを学習した。表 2.1に実験条件を示す。

解析の戦略も極めて単純なものにした。句構造解析モジュールはボトムアップ探索を行なうチャートパーザなので、とりあえずチャートの弧の数が上限 (現在は 20000) に達するまで全解探索を行ない、入力が適格な文として解釈された場合には、文としての全ての解釈を結果として出力した。入力が不適格な文であったり、適格な文でもメモリ不足になった場合には、チャートに保持されれいる部分木を、左最長優先のヒューリスティックを利用して探索し、最大 50 通りの部分木の組合せを結果として出力した。こうして出力した木 (または部分木の集合) のすべての中から、

1. たまたま最初に見つかった結果 (FIRST-HIT)

2. 通常の確率文法を用いてスコアリングし、もっとも高いスコアを得た結果 (PCFG)

文法規則 (核文法)	229 規則
テスト集合	2352 文 (64 会話)
最長	49 語
最短	2 語
平均	11.6 語
訓練集合	4237 文 (104 会話)

Table 2.1: 実験条件

	recall	precision	crossing
FIRST HIT	88.2%	88.8%	9.2%
PCFG	89.4%	90.5%	8.2%
RULE-BIGRAM	92.1%	92.9%	6.1%

Table 2.2: 実験結果

3. 北により提案された言語モデルを用いてスコアリングし、もっとも高いスコアを得た結果 (RULE-BIGRAM)

の 3 つの解を求め、評価した。なお解の評価は、Black[7] により提案された手法で行なった。表 2.2に実験結果を示す。

今回の実験では、あらかじめ正しく形態素解析された入力という、現実にはあり得ない入力を用いているので、結果の数値の絶対値には意味がほとんどない。しかし、ある程度まとまった量の解析実験なので、北の言語モデル (RULE-BIGRAM) は曖昧性解消のための知識としてかなり優れている、と判断していいだろう。

2.4 おわりに

本稿では、現在開発を進めている構文解析ツールキットの概要と、ツールキットを利用した簡単な解析実験の結果を報告した。複雑で難解な構文・意味解析機構をできる限り単純なモジュールの組合せで実現することにより、様々な言語知識を有効に利用したり、外部モジュールとのリンクが容易になることが期待される。

今後は、開発途中の各モジュールを完成させるとともに、

- 分割された各モジュールをどのように統合して利用するか。単純な階層型インターフェー

スでよいのか、あるいは別のインターフェースを研究する必要があるのか。

- 分散して管理されることになる言語知識の一貫性をどう保つか。
- 音声認識部とのインターフェースをどうするか。

等の検討を行ない、より頑健かつ高精度で処理効率が良い構文・意味解析機構を目指していく。

形態素解析モジュール

基本機能	文字列から形態素列への変換
データ構造変換知識	辞書 (形態素辞書)
曖昧性解消知識	単語の N-gram、品詞 (ラベル) の N-gram、接続テーブル 、最長一致等 のヒューリスティック等

形態素調整モジュール

基本機能	形態素列のデータ書き換え
データ書き換え知識	形態素列書き換え規則
曖昧性解消知識	単語の N-gram、品詞 (ラベル) の N-gram、接続テーブル 、最長一致等 のヒューリスティック等

句構造解析モジュール

基本機能	形態素列から句構造への変換
データ構造変換知識	句構造規則 (文脈自由文法)
曖昧性解消知識	確率文法、規則の連鎖統計情報、 語や句の共起関係、 Mental OS 等のヒューリスティック等

構文木調整モジュール

基本機能	句構造のデータ書き換え
データ書き換え知識	句構造書き換え規則
曖昧性解消知識	確率文法、規則の連鎖統計情報、 語や句の共起関係、 Mental OS 等のヒューリスティック等

依存構造解析モジュール

基本機能	句構造から依存構造への変換
データ構造変換知識	注釈付き句構造規則
曖昧性解消知識	規則の適用確率、 語や句の共起関係、統語的な制約、 枝分かれ等に関するヒューリスティック等

格構造解析モジュール

基本機能	依存構造 (ラベルなしグラフ) から格構造 (ラベル付きグラフ) への変換
データ構造変換知識	格情報辞書
曖昧性解消知識	意味素性、規則の適用確率、 語や句の共起関係、シソーラス 、 ドメインに依存するヒューリスティック等

意味解析モジュール

基本機能	ラベル付きグラフの書き換え
データ構造変換知識	書き換え規則
曖昧性解消知識	規則の適用確率, メタ規則等

Figure 2.3: 構文解析ツールキットの各モジュールの概要

入力:
ニューワシントンホテルでございます。

出力:
((ニューワシントンホテル 固有名詞)(で 助動詞)
(ございま 補助動詞)(す 語尾)(。 記号))

Figure 2.4: 形態素解析モジュールの入出力例

入力:
((ニューワシントンホテル 固有名詞)(で 助動詞)
(ございま 補助動詞)(す 語尾)(。 記号))

出力:
((ニューワシントンホテル <固有名詞>)(で <助動詞>)
(ございま <補助動詞>)(す <語尾>)(。 <記号>))

Figure 2.5: 形態素調整モジュールの入出力例

入力:
((ニューワシントンホテル <固有名詞>)(で <助動詞>)
(ございま <補助動詞>)(す <語尾>)(。 <記号>))

出力:
```
(<文>
  (<節>
    (<動詞句>
      (<動詞>
        (<名詞句>
          (<固有名詞> ニューワシントンホテル))
        (<助動詞> で))
      (<補助動詞>
        (<補助動詞語幹> ございま)
        (<語尾> す))))
  (<句点> 。))
```

Figure 2.6: 句構造解析モジュールの入出力例

入力:

```
(文
 (主語文節 ((ニューワシントンホテル <固有名詞>)
            (で <助動詞>)))
 (補語文節 ((ございま <補助動詞>)(す <語尾>)
            (。 <記号>))))
```

出力:

```
(<文>
 (<節>
  (<動詞句>
   (<動詞>
    (<名詞句>
     (<固有名詞> ニューワシントンホテル))
    (<助動詞> で))
   (<補助動詞>
    (<補助動詞語幹> ございま)
    (<語尾> す))))
 (<句点> 。))
```

※文節文法から一般の句構造規則への調整結果を例として示す。

Figure 2.7: 構文木調整モジュールの入出力例

入力:

```
(<文>
 (<節>
  (<動詞句>
   (<動詞>
    (<名詞句>
     (<固有名詞> ニューワシントンホテル))
    (<助動詞> で))
   (<補助動詞>
    (<補助動詞語幹> ございま)
    (<語尾> す))))
 (<句点> 。))
```

出力:

```
(。):(<句点>)
└┬(ございま す):(<補助動詞語幹> <語尾>)
 └┬(で):(<助動詞>)
  └─(ニューワシントンホテル):(<固有名詞>)
```

Figure 2.8: 依存構造解析モジュールの入出力例

```
入力:
  (。):(<句点>)
  └┬(ございます):(<補助動詞語幹> <語尾>)
   └┬(で):(<助動詞>)
    └─(ニューワシントンホテル):(<固有名詞>)

出力:
[ [語義見出し 。]
  [カテゴリ <句点>]
  [任意 [ [語義見出し *ございます*]
          [カテゴリ <補助動詞語幹>]
          [任意 [ [語義見出し *です*]
                  [カテゴリ <助動詞>]
                  [OBJE *未定義*]
                  [IDEN
                   [ [語義見出し
                      *ニューワシントンホテル*]
                     [カテゴリ
                      <固有名詞>] ] ] ] ] ] ] ]
```

Figure 2.9: 格構造解析モジュールの入出力例

```
入力:
[ [語義見出し 。]
  [カテゴリ <句点>]
  [任意 [ [語義見出し *ございます*]
          [カテゴリ <補助動詞語幹>]
          [任意 [ [語義見出し *です*]
                  [カテゴリ <助動詞>]
                  [OBJE *未定義*]
                  [IDEN
                   [ [語義見出し
                      *ニューワシントンホテル*]
                     [カテゴリ
                      <固有名詞>] ] ] ] ] ] ] ]

出力:
[ [RELN *INFORM*]
  [AGEN *SPEAKER*]
  [RECP *HEARER*]
  [OBJE [ [RELN *COPULA*]
          [OBJE
           *UNSPECIFIED-COMPLEX*]
          [IDEN
           [ [RELN *ニューワシントンホテル*] ] ] ] ] ]
```

Figure 2.10: 意味解析モジュールの入出力例

```
(
    (＜感動詞＞
        (＜副詞句＞
            (＜副詞＞　大変))
        (＜感動詞＞
            (＜感動詞＞　申し訳ございません)
            (＜読点＞　、)))
    (＜名詞句＞
        (＜人名＞　鈴木)
        (＜接尾辞＞　様))
    (＜句点＞　。)
)
```

Figure 2.11: 不適格文の部分木による表現

参考文献

[1] Jensen, K. and Heidorn, G.E: "The Fitted Parsing : 100% Parsing Capability in a Syntactic Grammar of English," ANLP83, 1983.

[2] Kita, K et al.: "Continuously Spoken Sentence Recognition by HMM-LR," ICSLP-92, pp.305-308, 1992.

[3] 長尾 確, 丸山 宏: "自然言語処理における曖昧さとその解消, 情報処理," Vol.33, No. 7, 1992.

[4] Nagata, M. and Morimoto, T.: "A Unification-Based Japanese Parser for Speech-to-Speech Translation," IEICE Trans. Inf. & Syst., Vol.E76-D, No.1, pp.51-61 1993.

[5] Tashiro, T., Uratani, N., Morimoto, T, "Restructuring Tagged Corpora with Morpheme Adjustment Rules", COLING94, 1994.

[6] 田代敏久 , 柏岡 秀紀 , Ezra W.Black, "構文木コーパスの再構成手法" , 情報処理学会第49 回全国大会, 1994.

[7] Black, E., et al.: "A Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars", DARPA Speech and Natural Language Workshop, 1991.

[8] Morimoto, T. et al. : "A Speech and Language Database for Speech Translation Research", ICSLP94, 1994.